PROCEEDINGS A

royalsocietypublishing.org/journal/rspa

Research

Cite this article: Kaheman K, Kutz JN, Brunton SL. 2020 SINDy-PI: a robust algorithm for parallel implicit sparse identification of nonlinear dynamics. *Proc. R. Soc. A* **476**: 20200279.
http://dx.doi.org/10.1098/rspa.2020.0279

Received: 14 April 2020
Accepted: 10 September 2020

Subject Areas:
applied mathematics, differential equations

Keywords:
system identification, optimization, model selection, rational differential equations

Author for correspondence:
Kadierdan Kaheman
e-mail: kadierk@uw.edu

THE ROYAL SOCIETY
PUBLISHING

SINDy-PI: a robust algorithm for parallel implicit sparse identification of nonlinear dynamics

Kadierdan Kaheman[1], J. Nathan Kutz[2] and Steven L. Brunton[1]

[1]Department of Mechanical Engineering, and [2]Department of Applied Mathematics, University of Washington, Seattle, WA 98195, USA

KK, 0000-0003-2279-2793

Accurately modelling the nonlinear dynamics of a system from measurement data is a challenging yet vital topic. The sparse identification of nonlinear dynamics (SINDy) algorithm is one approach to discover dynamical systems models from data. Although extensions have been developed to identify implicit dynamics, or dynamics described by rational functions, these extensions are extremely sensitive to noise. In this work, we develop SINDy-PI (parallel, implicit), a robust variant of the SINDy algorithm to identify implicit dynamics and rational nonlinearities. The SINDy-PI framework includes multiple optimization algorithms and a principled approach to model selection. We demonstrate the ability of this algorithm to learn implicit ordinary and partial differential equations and conservation laws from limited and noisy data. In particular, we show that the proposed approach is several orders of magnitude more noise robust than previous approaches, and may be used to identify a class of ODE and PDE dynamics that were previously unattainable with SINDy, including for the double pendulum dynamics and simplified model for the Belousov–Zhabotinsky (BZ) reaction.

1. Introduction

Discovering dynamical system models from data is critically important across science and engineering.

Traditionally, models are derived from first principles, although this approach may be prohibitively challenging in many fields, such as climate science, finance and biology. Fortunately, data-driven model discovery (i.e. system identification) is a rapidly developing field [1], with a range of techniques including classic linear approaches [2,3], dynamic mode decomposition (DMD) [4,5] and Koopman theory [6–9], nonlinear autoregressive models [10,11], neural networks [12–20], Gaussian process regression [21,22], nonlinear Laplacian spectral analysis [23], diffusion maps [24], genetic programming [25–27], and sparse regression [28–30], to highlight some of the recent developments. Of particular note is a recent push towards *parsimonious* modelling [25,26,28], which favours Pareto-optimal models with the lowest complexity required to describe the observed data. These models benefit from being interpretable, and they tend to generalize and prevent overfitting. The sparse identification of nonlinear dynamics (SINDy) algorithm [28] discovers parsimonious models through a sparsity-promoting optimization to select only a few model terms from a library of candidate functions. SINDy has been widely adopted in the community [30–46], but it relies on the dynamics having a sparse representation in a pre-defined library, making it difficult to discover implicit dynamics and rational functions. The implicit-SINDy extension [47] makes it possible to identify these implicit functions, although this algorithm is extremely sensitive to noise. In this work, we develop a robust, parallel algorithm for the sparse identification of implicit dynamics, making it possible to explore entirely new classes of systems that were previously inaccessible.

Parsimonious modelling has a rich history, with many scientific advances being argued on the basis of Occam's razor, that the simplest model is likely the correct one. SINDy exemplifies this principle, identifying a potentially nonlinear model with the fewest terms required to describe how the measurement data changes in time. The basic idea behind SINDy may be illustrated on a one-dimensional system $\dot{x} = f(x)$; the general formulation for multidimensional dynamics will be described in the following sections. An interpretable form of the nonlinear dynamics may be learned by writing the rate of change of the state of the system x as a sparse linear combination of a few terms in a library of candidate functions, $\boldsymbol{\Theta}(x) = [\,\theta_1(x)\ \theta_2(x)\ \dots\ \theta_p(x)\,]$:

$$\dot{x}(t) = f(x(t)) \approx \boldsymbol{\Theta}(x(t))\boldsymbol{\xi}, \tag{1.1}$$

where each $\theta_j(x)$ is prescribed candidate term (e.g. $x, x^2, \sin(x), \dots$). The derivative of the state and the library of candidate functions may both be computed from measured trajectory data. It then remains to solve for a sparse vector $\boldsymbol{\xi}$ with non-zero entries ξ_j indicating which functions $\theta_j(x)$ are active in characterizing the dynamics. The resulting models strike a balance between accuracy and efficiency, and they are highly interpretable by construction. In a short time, the SINDy algorithm has been extended to include inputs and control [48], to identify partial differential equations [29,30], to incorporate physically relevant constraints [34], to include tensor bases [45], and to incorporate integral terms for denoising [49,50]. These extensions and its simple formulation in terms of a generalized linear model in (1.1) have resulted in SINDy being adopted in the fields of fluid mechanics [34,37], nonlinear optics [31], plasma physics [32], chemical reactions [33,36,39], numerical methods [41] and structural modelling [42].

The generalized linear model in (1.1) does not readily lend itself to representing implicit dynamics and rational functions, which are not naturally expressible as sum of a few basis functions. Instead, the implicit-SINDy algorithm [47] reformulates the SINDy problem in an implicit form:

$$\boldsymbol{\Theta}(x, \dot{x})\boldsymbol{\xi} = 0. \tag{1.2}$$

This formulation is flexible enough to handle a much broader class of dynamics with rational function nonlinearities, such as $\dot{x} = N(x)/D(x)$ which may be rewritten as $\dot{x}D(x) + N(x) = 0$. However, the sparsest vector $\boldsymbol{\xi}$ that satisfies (1.2) is the trivial solution $\boldsymbol{\xi} = \mathbf{0}$. Thus, the implicit-SINDy algorithm leverages a recent non-convex optimization procedure [51,52] to find the sparsest vector $\boldsymbol{\xi}$ in the null space of $\boldsymbol{\Theta}(x, \dot{x})$, which differs from other approaches [53,54] that identify the rational dynamics. For even small amounts of noise, the dimension of the null space will become prohibitively large, making this approach extremely sensitive to noise and compromising the model discovery process.

This work develops an optimization and model selection framework that recasts implicit-SINDy as a convex problem, making it as noise robust as the original non-implicit SINDy algorithm and enabling the identification of implicit ODEs and PDEs that were previously inaccessible. The key to making the implicit-SINDy algorithm robust is the realization that if we know even a single term in the dynamics, corresponding to a non-zero entry ξ_j, then we can rewrite (1.2) in a non-implicit form

$$\theta_j(x, \dot{x}) = \boldsymbol{\Theta}'(x, \dot{x})\boldsymbol{\xi}', \tag{1.3}$$

where $\boldsymbol{\Theta}'$ and $\boldsymbol{\xi}'$ have the jth element removed. Because none of these terms are known *a priori*, we sweep through the library, term by term, testing (1.3) for a sparse model that fits the data. This procedure is highly parallelizable and provides critical information for model selection. Our approach is related to the recent work of Zhang *et al.* [46], which also makes the implicit problem more robust by testing candidate functions individually. However, there are a number of key differences in the present approach. Our work explicitly considers *rational* nonlinearities to discover exceedingly complex implicit PDEs, such as a simplified model of the Belousov–Zhabotinsky (BZ) reaction. Our framework also provides several new greedy algorithms, including parallel and constrained formulations. We further extend this method to include the effect of control inputs, making it applicable to robotic systems [55], and we use this procedure to discover Hamiltonians. Finally, our approach provides guidance on model selection, a comprehensive comparison with previous methods, and a careful analysis of noise robustness.

2. Background

We briefly introduce the full multidimensional SINDy and implicit-SINDy algorithms, which will provide a foundation for our robust implicit identification algorithm in §3.

(a) Sparse identification of nonlinear dynamics

The goal of SINDy [28] is to discover a dynamical system

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x}(t) = \boldsymbol{f}(\boldsymbol{x}(t)), \tag{2.1}$$

from time-series data of the state $\boldsymbol{x}(t) = [x_1(t), \ldots, x_n(t)]^T \in \mathbb{R}^n$. We assume that the dynamics, encoded by the function $\boldsymbol{f}$, admit a sparse representation in a library of candidate functions:

$$\boldsymbol{\Theta}(\boldsymbol{x}) = \begin{bmatrix} \theta_1(\boldsymbol{x}) & \theta_2(\boldsymbol{x}) & \cdots & \theta_p(\boldsymbol{x}) \end{bmatrix}. \tag{2.2}$$

Thus, each row equation in (2.1) may be written as

$$\frac{\mathrm{d}}{\mathrm{d}t}x_k(t) = f_k(\boldsymbol{x}(t)) \approx \boldsymbol{\Theta}(\boldsymbol{x})\boldsymbol{\xi}_k, \tag{2.3}$$

where $\boldsymbol{\xi}_k$ is a sparse vector, indicating which terms are active in the dynamics.

We determine the non-zero entries of $\boldsymbol{\xi}_k$ through sparse regression based on trajectory data. The time-series data is arranged into a matrix $\mathbf{X} = [\, x(t_1)\ x(t_2)\ \cdots\ x(t_m)\,]^T$, and the associated time derivative matrix $\dot{\mathbf{X}} = [\, \dot{x}(t_1)\ \dot{x}(t_2)\ \cdots\ \dot{x}(t_m)\,]^T$ is computed using an appropriate numerical differentiation scheme [1,28,56]. It is then possible to evaluate the library $\boldsymbol{\Theta}$ on trajectory data in $\mathbf{X}$ so that each column of $\boldsymbol{\Theta}(\mathbf{X})$ is a function θ_j evaluated on the m snapshots in $\mathbf{X}$.

It is now possible to write the dynamical system in terms of a generalized linear model, evaluated on trajectory data:

$$\dot{\mathbf{X}} = \boldsymbol{\Theta}(\mathbf{X})\boldsymbol{\Xi}. \tag{2.4}$$

There are several approaches to identify the sparse matrix of coefficients $\boldsymbol{\Xi}$, including sequentially thresholded least squares (STLSQ) [28,57], LASSO [58], sparse relaxed regularized

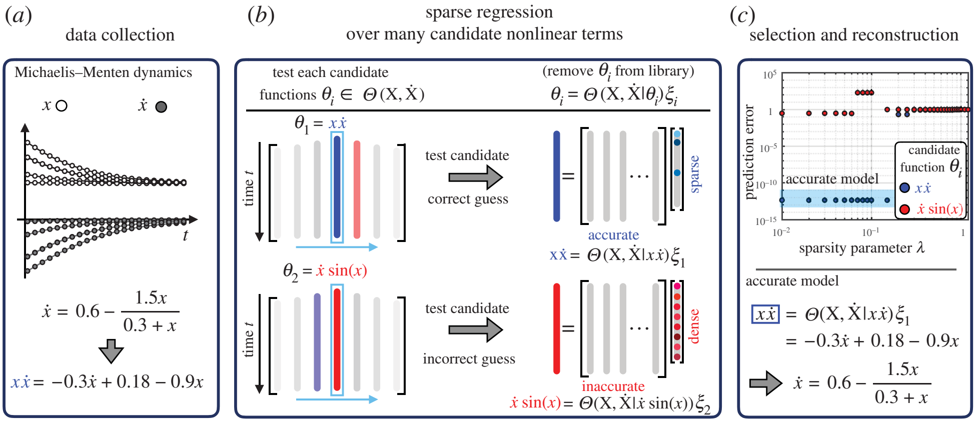

(a)
data collection
Michaelis–Menten dynamics
x ○
$\dot{x}$ ●
t
$\dot{x} = 0.6 - \frac{1.5x}{0.3+x}$
$x\dot{x} = -0.3\dot{x} + 0.18 - 0.9x$
(b)
sparse regression over many candidate nonlinear terms
test each candidate functions $\theta_i \in \Theta(\mathrm{X}, \dot{\mathrm{X}})$
(remove θ_i from library)
$\theta_i = \Theta(\mathrm{X}, \dot{\mathrm{X}}|\theta_i)\xi_i$
$\theta_1 = x\dot{x}$
time t
test candidate
correct guess
sparse
accurate
$x\dot{x} = \Theta(\mathrm{X}, \dot{\mathrm{X}}|x\dot{x})\xi_1$
$\theta_2 = \dot{x}\sin(x)$
time t
test candidate
incorrect guess
dense
inaccurate
$\dot{x}\sin(x) = \Theta(\mathrm{X}, \dot{\mathrm{X}}|\dot{x}\sin(x))\xi_2$
(c)
selection and reconstruction
prediction error
10^5
1
10^{-5}
10^{-10}
10^{-15}
10^{-2}
10^{-1}
1
sparsity parameter λ
accurate model
candidate function θ_i
● $x\dot{x}$
● $\dot{x}\sin(x)$
accurate model
$\boxed{x\dot{x}} = \Theta(\mathrm{X}, \dot{\mathrm{X}}|x\dot{x})\xi_1$
$= -0.3\dot{x} + 0.18 - 0.9x$
$\dot{x} = 0.6 - \frac{1.5x}{0.3+x}$

Figure 1. The illustration of the SINDy-PI algorithm on Michaelis–Menten dynamics. (*a*) The Michaelis–Menten system is simulated, and measurement data is provided to SINDy-PI. (*b*) Multiple possible left-hand side functions are tested at the same time. (*c*) The candidate model prediction error is calculated, and the best model is selected. (Online version in colour.)

regression (SR3) [59,60], stepwise sparse regression (SSR) [36] and Bayesian approaches [46,61]. It is possible to augment the library to include partial derivatives for the identification of partial differential equations (PDEs) [29,30]. Similarly, it is possible to include external forcing terms in the library $\boldsymbol{\Theta}$, enabling the identification of forced and actively controlled systems [48]. To alleviate the effect of noise, it is possible to reframe the SINDy problem in terms of an integral formulation [49,50]. There are a number of factors that affect the robustness of SINDy, some of which are discussed in appendix I.

(b) Implicit sparse identification of nonlinear dynamics

The implicit-SINDy algorithm [47] extends SINDy to identify implicit differential equations

$$f(x, \dot{x}) = 0, \tag{2.5}$$

and in particular, systems that include rational functions in the dynamics, such as chemical reactions and metabolic networks that have a separation of timescales.

The implicit-SINDy generalizes the library $\boldsymbol{\Theta}(\mathbf{X})$ in (2.4) to include functions of x and $\dot{x}$:

$$\boldsymbol{\Theta}(\mathbf{X}, \dot{\mathbf{X}})\boldsymbol{\Xi} = \mathbf{0}. \tag{2.6}$$

However, this approach requires solving for a matrix $\boldsymbol{\Xi}$ whose columns $\boldsymbol{\xi}_k$ are sparse vectors in the null space of $\boldsymbol{\Theta}(\mathbf{X}, \dot{\mathbf{X}})$. This approach is non-convex, relying on the alternating directions method (ADM) [47,52], and null space computations are highly ill-conditioned for noisy data [1,47,62], thus inspiring the current work and mathematical innovations.

3. SINDy-PI: robust parallel identification of implicit dynamics

We have developed the SINDy-PI (parallel, implicit) framework for the robust identification of implicit dynamics, bypassing the null space approach discussed in §2b (figure 1). The idea is that if even a single term $\theta_j(x, \dot{x}) \in \boldsymbol{\Theta}(x, \dot{x})$ in the dynamics (2.5) is known, it is possible to rewrite (2.6) as

$$\theta_j(\mathbf{X}, \dot{\mathbf{X}}) = \boldsymbol{\Theta}(\mathbf{X}, \dot{\mathbf{X}}|\theta_j(\mathbf{X}, \dot{\mathbf{X}}))\boldsymbol{\xi}_j, \tag{3.1}$$

where $\boldsymbol{\Theta}(\mathbf{X}, \dot{\mathbf{X}}|\theta_j(\mathbf{X}, \dot{\mathbf{X}}))$ is the library $\boldsymbol{\Theta}(\mathbf{X}, \dot{\mathbf{X}})$ with the θ_j column removed. equation (3.1) is no longer in implicit form, and the sparse coefficient matrix corresponding to the remaining terms may be solved for using previously developed SINDy techniques [28–30,36,46,49,50,59–61]. In particular, we solve for a sparse coefficient vector $\boldsymbol{\xi}_j$ that minimizes the following loss function:

$$\|\theta_j(\mathbf{X}, \dot{\mathbf{X}}) - \boldsymbol{\Theta}(\mathbf{X}, \dot{\mathbf{X}}|\theta_j(\mathbf{X}, \dot{\mathbf{X}})\boldsymbol{\xi}_j\|_2 + \beta \left\|\boldsymbol{\xi}_j\right\|_0, \tag{3.2}$$

where β is the sparsity promoting parameter. There are numerous relaxations of the non-convex optimization problem in (3.2), for example the sequentially thresholded least-squares algorithm [28]. Because there is no null space calculation, the resulting algorithm is considerably more robust to noise than the implicit-SINDy algorithm [47], i.e. we longer have to deal with an ill-conditioned null space problem.

In general, the entire point of SINDy is that the dynamics are not known ahead of time, and so it is necessary to test each candidate function θ_j until one of the models in (3.1) admits a sparse and accurate solution. When an incorrect candidate term is used, then the algorithm results in a dense (non-sparse) model $\boldsymbol{\xi}_j$ and an inaccurate model fit, and when a correct term is included, the algorithm identifies a sparse model $\boldsymbol{\xi}_j$ and an accurate model fit. In this way, it is clear when the algorithm has identified the correct model. Moreover, there is a wealth of redundant information, since each term in the correct model may be used as the candidate function on the left-hand side, and the resulting models may be cross-referenced. This approach is highly parallelizable, and each candidate term may be tested simultaneously in parallel. The non-parallel formulation in (3.1) was recently introduced by Zhang *et al.* [46] in the context of Bayesian regression, where they also make the implicit problem more robust by testing candidate functions individually; however, they do not consider dynamics with rational function nonlinearities or control inputs. In this work, we extend the robust implicit formulation to identify several challenging implicit ODE and PDE systems with rational function nonlinearities, which are ubiquitous in engineering and natural systems, and systems with external forcing and control inputs. We also introduce the parallel formulation and model selection frameworks. Further, we will introduce a constrained optimization framework to simultaneously test all candidate functions.

(a) Model selection

For each candidate function in (3.1), we obtain one candidate model. When the candidate function θ_j is not in the true dynamics, then the resulting coefficient vector $\boldsymbol{\xi}_j$ will not be sparse and there will be large prediction error. In contrast, when a correct candidate function is selected, then we obtain a sparse coefficient vector $\boldsymbol{\xi}_j$ and small prediction error. For an implicit dynamical system, there may be several different implicit equations that must be identified, resulting in several candidate functions that admit sparse models. The sequentially thresholded least-squares (STLSQ) algorithm that we use here, and whose convergence properties are considered by Zhang and Schaeffer [57], iteratively computes a least-squares solution to minimize $\|\theta_j(\mathbf{X}, \dot{\mathbf{X}}) - \boldsymbol{\Theta}(\mathbf{X}, \dot{\mathbf{X}}|\theta_j(\mathbf{X}, \dot{\mathbf{X}}))\boldsymbol{\xi}_j\|_2$ and then zeros out small entries in $\boldsymbol{\xi}_j$ that are below a set threshold λ. This threshold λ is a hyperparameter that must be tuned to select the model that most accurately balances accuracy and efficiency. Thus, we must employ model selection techniques to identify the implicit models that best supports the data, while remaining as simple as possible.

There are several valid approaches to model selection. To select a parsimonious yet accurate model we can also employ the Akaike information criterion (AIC) [63,64] and Bayesian information criterion (BIC) [65], as in [66]. It is also possible to sweep through the parameter λ and candidate functions θ_j, and then choose the Pareto optimal model from a *family* of models on the Pareto front balancing accuracy and efficiency; this is the approach in the original SINDy work [28] and in earlier work leveraging genetic programming to discover dynamics [25,26]. In this work, we take a different approach, selecting models based on performance on a test dataset $\mathbf{X}_t$ that has been withheld for model validation to automate the model selection process. For each threshold λ, the resulting model is validated on the test set $\mathbf{X}_t$, and the model with the lowest test error is selected. One error function is the model fit:

$$\text{Error} = \frac{\left\|\theta_j(\mathbf{X}_t, \dot{\mathbf{X}}_t) - \boldsymbol{\Theta}(\mathbf{X}_t, \dot{\mathbf{X}}_t | \theta_j(\mathbf{X}_t, \dot{\mathbf{X}}_t))\boldsymbol{\Xi}\right\|_2}{\left\|\theta_j(\mathbf{X}_t, \dot{\mathbf{X}}_t)\right\|_2}. \tag{3.3}$$

In practice, for rational dynamics, we select based upon the predicted derivative $\dot{\mathbf{X}}_t$:

$$\text{Error} = \frac{\left\|\dot{\mathbf{X}}_t - \dot{\mathbf{X}}_t^{\text{model}}\right\|_2}{\left\|\dot{\mathbf{X}}_t\right\|_2}. \tag{3.4}$$

For implicit dynamics where each state derivative may be written as a rational function

$$\dot{x}_k = f_k(\mathbf{x}) = \frac{N_k(\mathbf{x})}{D_k(\mathbf{x})}, \tag{3.5}$$

then we restrict the candidate functions to $\theta_j(\mathbf{x}, \dot{\mathbf{x}}) = \dot{x}_k\theta_j(\mathbf{x})$ for some $\theta_j(\mathbf{x}) \in \boldsymbol{\Theta}(\mathbf{x})$ to identify a separate sparse model for each $\dot{x}_k$. Several candidate functions may provide accurate and sparse models. These different models may further be cross-references to check that the same terms are being selected in each model, providing additional information for model selection and validation.

(b) Constrained optimization formulation

In (3.1) each candidate function was tested individually in a parallel optimization. However, each of these individual equations may be combined into a single constrained system of equations

$$\boldsymbol{\Theta}(\mathbf{X}, \dot{\mathbf{X}}) = \boldsymbol{\Theta}(\mathbf{X}, \dot{\mathbf{X}})\boldsymbol{\Xi} \quad \text{such that } \boldsymbol{\Xi}_{jj} = 0. \tag{3.6}$$

We constrain $\boldsymbol{\Xi}$ to have zero entries on the diagonal, as shown in figure 2, which is the same as removing the candidate function from the library in the separate optimization problems in (3.1). Without this constraint, the trivial solution $\boldsymbol{\Xi} = \mathbb{I}_{p\times p}$ will provide the sparsest $\boldsymbol{\Xi}$ and the most accurate model. This may be written as a formal constrained optimization problem:

$$\begin{aligned} \min_{\boldsymbol{\Xi}} \quad & ||\boldsymbol{\Theta}(\mathbf{X}, \dot{\mathbf{X}}) - \boldsymbol{\Theta}(\mathbf{X}, \dot{\mathbf{X}})\boldsymbol{\Xi}||_2 + \beta\|\boldsymbol{\Xi}\|_0, \\ \text{s.t.} \quad & \text{diag}(\boldsymbol{\Xi}) = \mathbf{0}. \end{aligned} \tag{3.7}$$

This optimization is non-convex, although there are many relaxations that result in accurate and efficient proxy solutions [28,58,59]. In this work, we will use sequentially thresholded least squares, so that any entry $\boldsymbol{\Xi}_{ij} < \lambda$ will be set to zero; the sparsity parameter λ is a hyperparameter, and each column equation may require a different parameter λ_j. The constrained formulation in (3.7) can be solved efficiently in modern optimization packages, and we use CVX [67,68]. After solving (3.7) we have numerous candidate models, one for each column $\boldsymbol{\xi}_k$ of $\boldsymbol{\Xi}$, given by

$$\boldsymbol{\Theta}(\mathbf{X}, \dot{\mathbf{X}})\boldsymbol{\xi}_j = 0. \tag{3.8}$$

The sparse models that result in an accurate fit are candidate implicit models, and they may be assessed using the model selection approaches outlined above. These various models may be cross-referenced for consistency, as the same models will have the same sparsity pattern. This information can then be used to refine the library $\boldsymbol{\Theta}$, for example to only include the non-zero entries in the sparse columns of $\boldsymbol{\Xi}$.

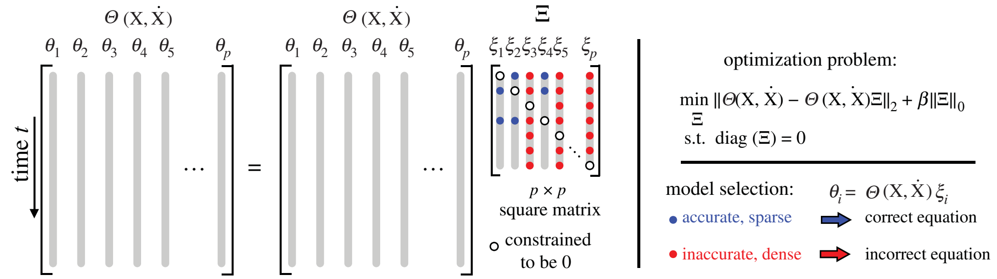


Figure 2. Schematic illustrating the constrained formulation of the SINDy-PI algorithm. (Online version in colour.)

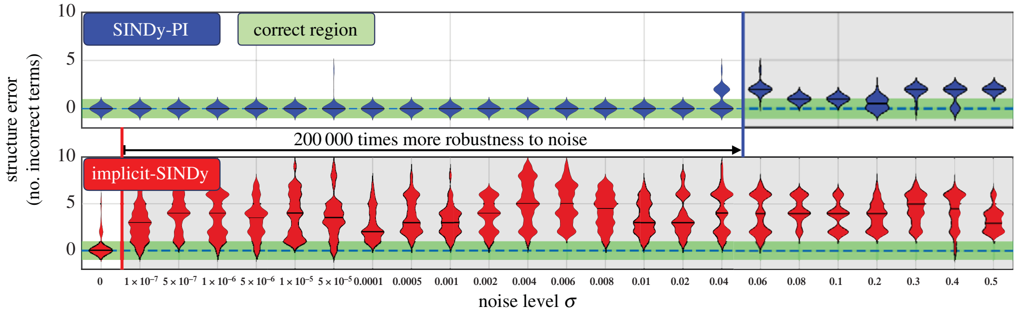


Figure 3. SINDy-PI and implicit-SINDy are compared on the Michaelis–Menten kinetics, where the structure error quantifies the number of terms in the model that are incorrectly added or deleted, compared with the true model. The derivative is computed by the total-variation regularization difference (TVRegDiff) [56] on noisy state measurements. The violin plots show the cross-validated distribution of the number of incorrect terms across 30 models. The green region (a rectangle stripe at zero value labelled as correct region) indicates no structural difference between the identified model and the ground truth model. Details are provided in appendix A(b). (Online version in colour.)

(c) Noise robustness

We now compare the noise sensitivity of SINDy-PI and implicit-SINDy on the one-dimensional Michaelis–Menten model for enzyme kinetics [47,69,70], given by

$$\dot{x} = j_x - \frac{V_{\max} x}{K_m + x}, \tag{3.9}$$

where x denotes the concentration of the substrate, j_x denotes the influx of the substrate, $V_{\max}$ denotes the maximum reaction time and K_m represents the concentration of half-maximal reaction. We use the same parameters as in [47], with $j_x = 0.6$, $V_{\max} = 1.5$, and $K_m = 0.3$. Figure 3 shows the result of the noise robustness of SINDy-PI and implicit-SINDy. In this example, SINDy-PI is able to handle over 10^5 more measurement noise than implicit-SINDy, while still accurately recovering the correct model. Details are provided in appendix A, and key factors that limit robustness are discussed in appendix I.

(d) Data usage

The data required to correctly identify a model is a critical aspect when comparing SINDy-PI and implicit-SINDy. Many experimental datasets are limited in volume, and thus our goal is to

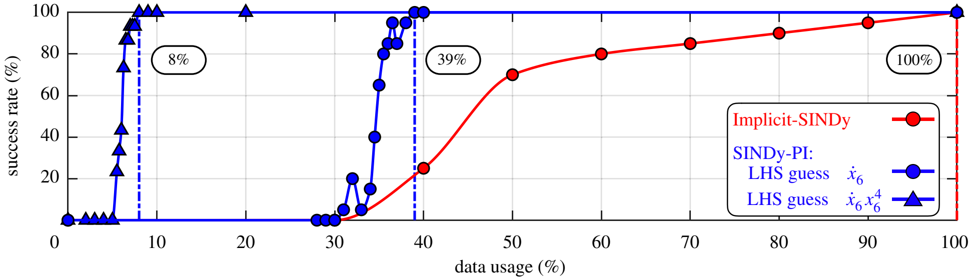


Figure 4. Success rate of SINDy-PI and implicit-SINDy identifying yeast glycolysis (3.10*f*) with different percentage of training data. Each data usage percentage is randomly sampled from the entire dataset composed of all trajectories. The success rate is calculated by averaging the results of 20 runs. (Online version in colour.)

identify a model with as little data as possible. In this section, we compare the SINDy-PI and implicit-SINDy methods on the challenging yeast glycolysis model [47,71] given by

$$\dot{x}_1 = c_1 + \frac{c_2 x_1 x_6}{1 + c_3 x_6^4}, \tag{3.10a}$$

$$\dot{x}_2 = \frac{d_1 x_1 x_6}{1 + d_2 x_6^4} + d_3 x_2 - d_4 x_2 x_7, \tag{3.10b}$$

$$\dot{x}_3 = e_1 x_2 + e_2 x_3 + e_3 x_2 x_7 + e_4 x_3 x_6 + f_5 x_4 x_7, \tag{3.10c}$$

$$\dot{x}_4 = f_1 x_3 + e_2 x_4 + f_3 x_5 + f_4 x_3 x_6 + f_5 x_4 x_7, \tag{3.10d}$$

$$\dot{x}_5 = g_1 x_1 + g_2 x_5, \tag{3.10e}$$

$$\dot{x}_6 = h_3 x_3 + h_5 x_6 + h_4 x_3 x_6 + \frac{h_1 x_1 x_6}{1 + h_2 x_6^4}, \tag{3.10f}$$

and

$$\dot{x}_7 = j_1 x_2 + j_2 x_2 x_7 + j_3 x_4 x_7. \tag{3.10g}$$

Equation (3.10*f*) is the most challenging equation to discover in this system, and figure 4 compares the success rate of SINDy-PI and implicit-SINDy in identifying this equation. SINDy-PI uses about 12 times less data than the implicit-SINDy when identifying (3.10*f*). Details are provided in appendices B and D.

(e) Comparison for implicit PDE identification

We now investigate the ability of SINDy-PI to discover a PDE with rational terms, given by a modified KdV equation

$$u_t = -u_{xxx} - 6uu_x - \gamma u + \frac{2g_0}{1+u}, \tag{3.11}$$

where γu is a loss term and $2g_0/(1+u)$ is a gain term. We fix $\gamma = 0.1$ and vary the value of g_0 from 0 to 1. As g_0 increases, the implicit term gradually dominates the dynamics. Figure 5 shows the results of SINDy-PI and PDE-FIND [29] for different values of g_0. For large g_0, SINDy-PI is able to accurately identify the rational function term, while this is not possible for PDE-FIND, since this term is not in the library. Details of the identification process are given in appendix C.

4. Advanced examples

We will now demonstrate the SINDy-PI framework on several challenging examples, including the double pendulum, an actuated single pendulum on a cart, the Belousov–Zhabotinsky PDE, and the identification of conserved quantities. All examples are characterized by rational nonlinearities, and we were unable to identify them using SINDy or implicit-SINDy, even in the absence of noise.

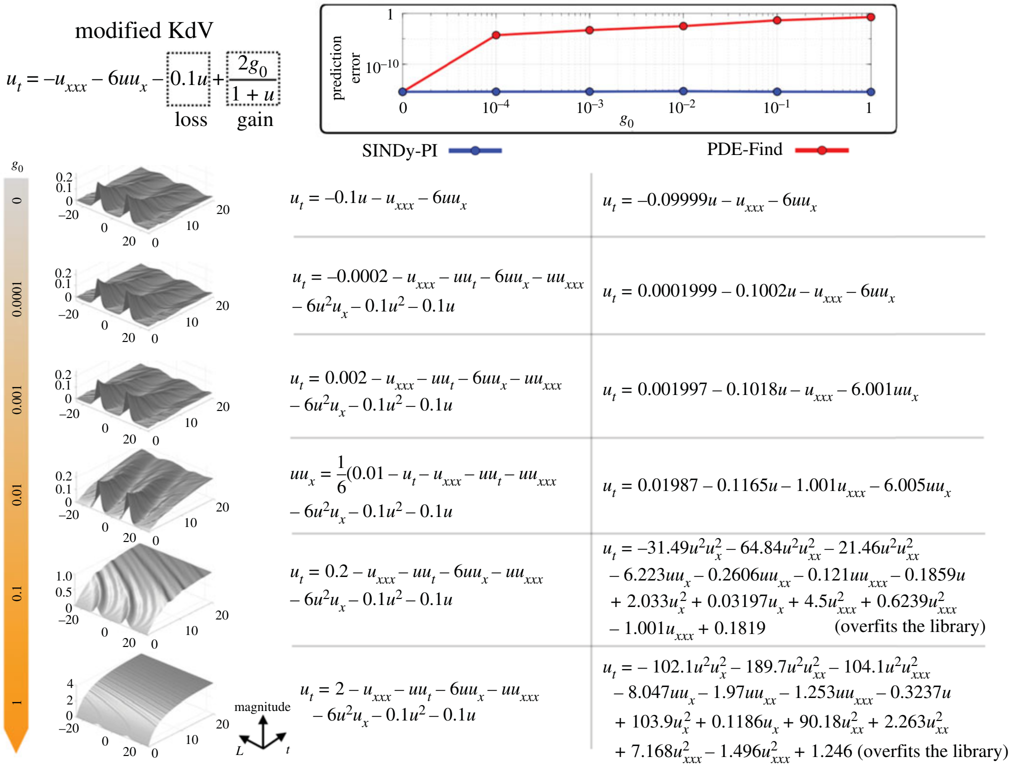


Figure 5. Comparison of SINDy-PI and PDE-FIND on an implicit PDE problem given by the modified KdV equation (3.11). As we increase g_0, the rational term begins to play a significant role in the system behaviour. For small g_0, PDE-FIND compensates for the effect of the rational term by tuning the other coefficients. When g_0 is large, PDE-FIND overfits the library. SINDy-PI, on the other hand, correctly identifies the rational term. (Online version in colour.)

(a) Mounted double pendulum

In our first example, we use SINDy-PI to discover the equations of motion of a mounted double pendulum, shown in figure 6. The double pendulum is a classic example of chaotic dynamics [72], and was an original challenging example used to demonstrate the capabilities of genetic programme for model discovery [26]. Correctly modelling the nonlinear dynamics is vital for accurate control [72].

We simulate the double pendulum dynamics, derived from the Euler–Lagrange equations, and use SINDy-PI to re-discover the dynamics from noisy measurements of the trajectory data. The governing equations and SINDy-PI models are provided in appendix F. Because these dynamics have rational nonlinearities, the original SINDy algorithm is unable to identify the dynamics, making this a challenging test case. The state vector is given by $\boldsymbol{x} = [\phi_1, \phi_2, \dot{\phi}_1, \dot{\phi}_2]^T$, and the parameters of the simulation are given in appendix D. The training data is generated from an initial condition $x_{\text{train}} = [\pi + 1.2,\ \pi - 0.6,\ 0,\ 0]^T$, simulated for 10 s using a time step of $\mathrm{d}t = 0.001$ s. The validation data is generated from an initial condition $x_{\text{val}} = [\pi - 1,\ \pi - 0.4,\ 0.3,\ 0.4]^T$, simulated for 3 s with time step $\mathrm{d}t = 0.001$ s.

To test the robustness of SINDy-PI, we add Gaussian noise to both the training and validation data. We test the resulting models using a new testing initial condition $x_{\text{test}} = [\pi + 0.3,\ \pi - 0.5,\ 0,\ 0]^T$. We construct our library $\boldsymbol{\Theta}$ to include over 40 trigonometric and polynomial terms. The most challenging part of this example is building a library with the necessary terms, without it growing too large. The library cannot be too extensive, or else the matrix $\boldsymbol{\Theta}$ becomes ill conditioned, making it sensitive to noise. To reduce the library size, we use one piece of expert

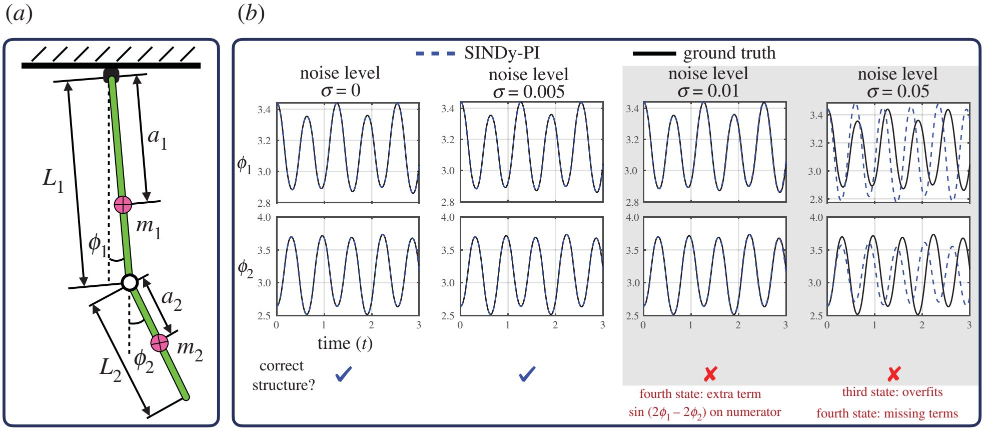


Figure 6. Schematic illustration of SINDy-PI identifying a mounted double pendulum system. (*a*) Data generation; (*b*) SINDy-PI Identified Model Performance. (Online version in colour.)

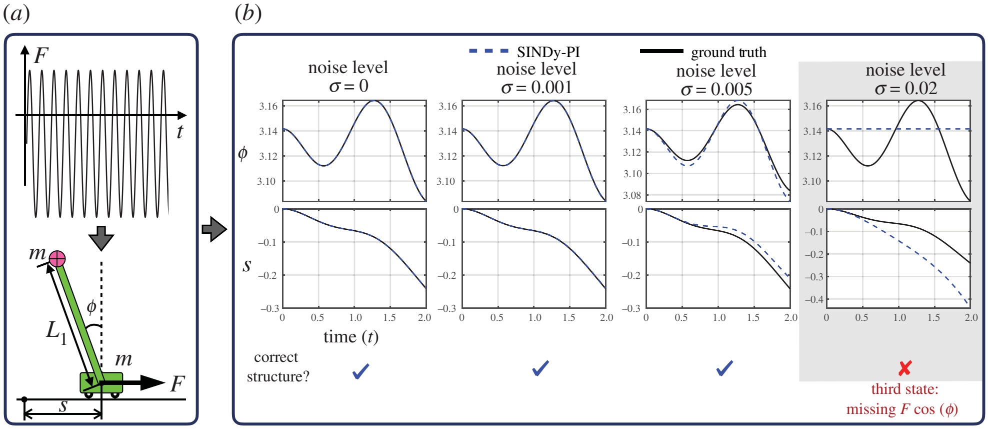


Figure 7. SINDy-PI is used to identify the single pendulum on a cart system. Control is applied to the cart, and both the cart and pendulum states are measured. When the measurement noise is small, SINDy-PI can identify the correct structure of the model. (*a*) Data generation; (*b*) SINDy-PI identified model performance. (Online version in colour.)

knowledge: the trigonometric terms should only consist of ϕ_1 and ϕ_2, the rotational angles of the pendula.

The candidate functions are chosen as a combination of state derivatives and trigonometric functions. Figure 6 shows that SINDy-PI can identify the equations of motion for low noise. For larger noise, SINDy-PI misidentifies the dynamics, although it still has short term prediction ability.

(b) Single pendulum on a cart

We now apply SINDy-PI to identify a fractional ODE problem with control input, given by the single pendulum on a cart in figure 7. SINDy has already been extended to include control inputs [48], although the original formulation doesn't accommodate rational functions.

The dynamics are derived from the Euler–Lagrange equations. All system parameters except for gravity are chosen to be 1, as summarized in appendix D; the governing equations and SINDy-PI models are shown in appendix E. The cart position is denoted by s. The state vector is given by

$\boldsymbol{x} = [\phi, s, \dot{\phi}, \dot{s}]^T$. The equations of motion are given by

$$\frac{\mathrm{d}}{\mathrm{d}t}\phi = \dot{\phi}, \tag{4.1a}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}s = \dot{s}, \tag{4.1b}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\dot{\phi} = -\frac{(M+m)g\sin(\phi) + FL_1\cos(\phi) + mL_1^2\sin(\phi)\cos(\phi)\dot{\phi}^2}{L_1^2(M+m-m\cos(\phi)^2)}, \tag{4.1c}$$

and
$$\frac{\mathrm{d}}{\mathrm{d}t}\dot{s} = \frac{mL_1^2\sin(\phi)\dot{\phi}^2 + FL_1 + mg\sin(\phi)\cos(\phi)}{L_1(M+m-m\cos(\phi)^2)}. \tag{4.1d}$$

Equation (4.1) is simulated with a time step of $\mathrm{d}t = 0.001$ to generate the training and testing data for model selection. The training data is generated using an initial condition $x_{\mathrm{train}} = [0.3,\ 0,\ 1,\ 0]^T$ with the control input chosen as $F_{\mathrm{train}} = -0.2 + 0.5\sin(6t)$, for time $t = 0$ to $t = 16$. Similarly, the validation data is generated using an initial condition $x_{\mathrm{val}} = [0.1,\ 0,\ 0.1,\ 0]^T$ with the control input chosen as $F_{\mathrm{val}} = -1 + \sin(t) + 3\sin(2t)$, for time $t = 0$ to $t = 2$.

The library is constructed using a combination of trigonometric and polynomial terms. Around 50 different basis functions are used for the library, and around 10 terms are tested as candidate functions. We add Gaussian noise to all system states. We then test the SINDy-PI model on a testing initial condition $x_{\mathrm{test}} = [\pi,\ 0,\ 0,\ 0]^T$ with control input $F_{\mathrm{test}} = -0.5 + 0.2\sin(t) + 0.3\sin(2t)$ for time $t = 0$ to $t = 2$. Figure 7 shows the resulting SINDy-PI models. The structure of the model is correctly identified up to a noise magnitude of 0.01. Beyond this noise level, the SINDy-PI identified model only has short term prediction ability.

(c) Simplified model of the Belousov–Zhabotinsky reaction

We now apply SINDy-PI to a challenging PDE with rational nonlinearities, a simplified model of the Belousov–Zhabotinsky (BZ) reaction. The simplified BZ reaction model is given by [73]

$$\frac{\partial x}{\partial \tau} = \frac{1}{\varepsilon}\left(\frac{fz(q-x)}{q+x} + x - x^2 - \beta x + s\right) + \frac{D_x}{D_u}\Delta x, \tag{4.2a}$$

$$\frac{\partial z}{\partial \tau} = x - z - \alpha z + \gamma u + \frac{D_z}{D_u}\Delta z, \tag{4.2b}$$

$$\frac{\partial s}{\partial \tau} = \frac{1}{\varepsilon_2}(\beta x - s + \chi u) + \frac{D_s}{D_u}\Delta s, \tag{4.2c}$$

and
$$\frac{\partial u}{\partial \tau} = \frac{1}{\varepsilon_3}\left[\alpha z - \left(\gamma + \frac{\chi}{2}\right)u\right] + \frac{D_u}{D_u}\Delta u, \tag{4.2d}$$

where x, z, s and u are dimensionless variables and $\Delta = (\partial^2/\partial x_s^2) + (\partial^2/\partial y_s^2)$ denotes the Laplacian operator.

The strong coupling dynamics and implicit behaviour in (4.2a) make the data-driven discovery of the simplified BZ reaction challenging when using implicit-SINDy and PDE-FIND. However, SINDy-PI correctly identifies the simplified dynamics of the BZ-Reaction, as shown in figure 8. To generate the simplified BZ reaction data, we use a spectral method [74,75] with time horizon $T = 1$ and time step of $\mathrm{d}t = 0.001$. We use $n = 128$ discretization points with spatial domain ranging from -10 to 10. The initial condition is chosen to be a mixture of Gaussian functions. Eighty per cent of the data is used for training, and the remaining 20% is used for model selection. The right-hand side library is normalized during the sparse regression process. A range of sparsity parameters λ are tested from 0.1 to 1, with increments of 0.1 The other system parameters in (4.2) are given in appendix D and the SINDy-PI model is given in appendix G.

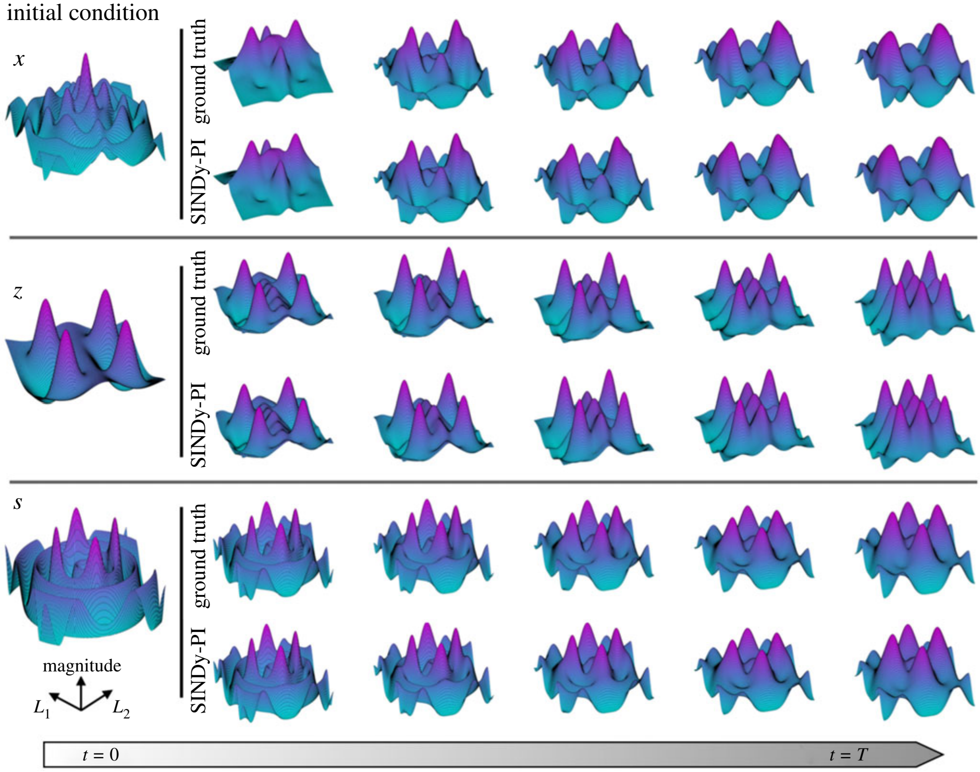
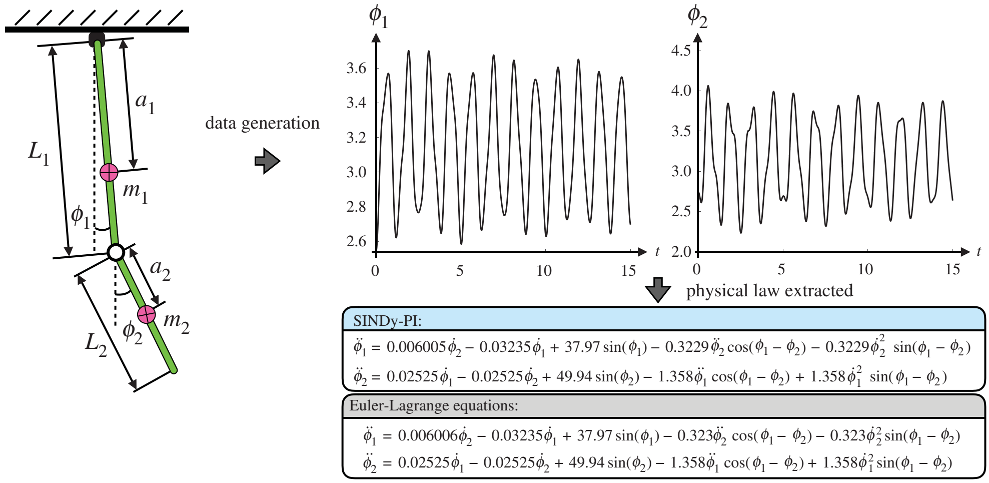


Figure 8. SINDy-PI is able to identify the simplified Belousov–Zhabotinsky reaction model. (Online version in colour.)



SINDy-PI:

$$\ddot{\phi}_1 = 0.006005\dot{\phi}_2 - 0.03235\dot{\phi}_1 + 37.97\sin(\phi_1) - 0.3229\ddot{\phi}_2\cos(\phi_1 - \phi_2) - 0.3229\dot{\phi}_2^2\sin(\phi_1 - \phi_2)$$

$$\ddot{\phi}_2 = 0.02525\dot{\phi}_1 - 0.02525\dot{\phi}_2 + 49.94\sin(\phi_2) - 1.358\ddot{\phi}_1\cos(\phi_1 - \phi_2) + 1.358\dot{\phi}_1^2\sin(\phi_1 - \phi_2)$$

Euler-Lagrange equations:

$$\ddot{\phi}_1 = 0.006006\dot{\phi}_2 - 0.03235\dot{\phi}_1 + 37.97\sin(\phi_1) - 0.323\ddot{\phi}_2\cos(\phi_1 - \phi_2) - 0.323\dot{\phi}_2^2\sin(\phi_1 - \phi_2)$$

$$\ddot{\phi}_2 = 0.02525\dot{\phi}_1 - 0.02525\dot{\phi}_2 + 49.94\sin(\phi_2) - 1.358\ddot{\phi}_1\cos(\phi_1 - \phi_2) + 1.358\dot{\phi}_1^2\sin(\phi_1 - \phi_2)$$

Figure 9. SINDy-PI is used to extract the conserved quantity for a double pendulum. (Online version in colour.)

(d) Extracting physical laws and conserved quantities

In this final example, we demonstrate how to use SINDy-PI to extract governing physical laws and conserved quantities from data. Many systems of interest are governed by Hamiltonian or Lagrangian dynamics. Instead of identifying the ODE or PDE equations of motion, it might be possible to extract the physical laws directly. These equations contain important information about the system and may be more concise, useful and straightforward than the underlying ODE or PDE. For example, given a Lagrangian, we can derive the equations of motion.

The most difficult aspect of using SINDy-PI to identify a physical law is how to build the library. Conservation laws may contain higher-order derivatives, such as $\ddot{x}$. To include all possible terms, the library may become exceedingly large. The library size will also increase if the system has many states. Large libraries make the sparse regression sensitive to noise. Thus, extracting the physical law from data using SINDy-PI is still challenging due to the lack of constraints when constructing the library function. We only show one example in our paper to demonstrate that it is possible to achieve this using SINDy-PI, but further work is required to reduce the library size so that the sparse regression is robust.

As an example, we consider the double pendulum shown in figure 9, with the system parameters given in appendix D. In this case, we also account for the friction in the pendulum joint, with friction constants of $k_1 = 7.2484 \times 10^{-4}$ and $k_2 = 1.6522 \times 10^{-4}$ for the pendulum arms, respectively. In this case, we extract the Lagrangian of the double pendulum [72] using SINDy-PI. To extract this Lagrangian, we simulate the system with initial condition $x_{\text{train}} = [\pi - 0.6,\ \pi - 0.4,\ 0,\ 0]^T$ from $t = 0$ to $t = 15$ with time step $\mathrm{d}t = 0.001$. The resulting model is shown in figure 9.

5. Conclusion and future work

In this paper, we develop SINDy-PI (parallel,implicit), a robust variant of the SINDy algorithm to identify implicit dynamics and rational nonlinearities. SINDy-PI overcomes the sensitivity of the previous implicit-SINDy approach, which is based on a null-space calculation, making it highly sensitive to noise. Instead, we introduce both parallel and constrained optimizations to test candidate terms in the dynamics, making the new SINDy-PI algorithm as robust as the original SINDy algorithm. We also extend the algorithm to incorporate external forcing and actuation, making it more applicable to real-world systems. We demonstrate this approach on several challenging systems with implicit and rational dynamics, including ODEs, actuated systems and PDEs. In particular, we discover the implicit dynamics for a simplified model for the BZ chemical reaction PDE, the double pendulum mechanical system and the yeast glycolysis model, which have all been challenging test cases for advanced identification techniques. Throughout these examples, we demonstrate considerable noise robustness and reductions to the data required, over the previous implicit-SINDy algorithm.

Despite the advances outlined here, there are still many important avenues of future work. One limitation of this approach, and of SINDy in general, is in the design of the library of candidate functions. The goal is a descriptive library, but the library size grows rapidly, which in turn makes the sparse regression ill-conditioned; other issues effecting robustness are discussed in appendix I. Recently, tensor approaches have been introduced to alleviate this issue, making libraries both descriptive and tractable [45], and this is a promising approach that may be incorporated in SINDy-PI as well. More generally, automatic library generation, guided by expert knowledge, is an important topic. Other research directions will involve parametrizing elements of the library, so that the algorithm simultaneously identifies the model structure and the parameters of the sparsely selected terms. Recent unified optimization frameworks, such as SR3 [59,60], may make this possible. Model selection is another key area that will required focused attention. Balancing accuracy on test data, sparsity of the model, and the potential for overfitting are all serious concerns. The sparse regression and optimization may also be improved for better noise robustness. Finally, modifying SINDy-PI to incorporate prior physical knowledge and to only model the discrepancy with an existing model [76] will be the focus of ongoing work.

Data accessibility. Our code and data could be seen in the following link: https://github.com/dynamicslab/SINDy-PI.

Authors' contributions. K.K. performed the research, designed the code and generated data and figures. All authors analysed the data, refined the figures and wrote the paper. All authors gave final approval for publication and agree to be held accountable for the work performed therein.

Competing interests. We declare we have no competing interest.

Funding. S.L.B. acknowledges support from the Army Research Office (grant no. ARO W911NF-19-1-0045) and the Air Force Office of Scientific Research (grant no. AFOSR FA9550-18-1-0200). J.N.K. acknowledges support from the Air Force Office of Scientific Research (grant no. AFOSR FA9550-17-1-0329).

Acknowledgements. We also acknowledge valuable discussions with Aditya Nair, Eurika Kaiser, Brian DeSilva, Tony Piaskowy, Jared Callaham and Benjamin Herrmann. We thank Ariana Mendible for reviewing the manuscript and providing useful suggestions.

Appendix A. Noise sensitivity of SINDy-PI and implicit-SINDy

(a) Performance evaluation criteria

To compare the performance of SINDy-PI and implicit-SINDy for noisy data, we must define an evaluation criteria. We compare the performance of the best model generated by each method that has the lowest prediction error on the test data, selected according to equation (3.4). To compare the models generated by the two methods with the ground truth model, we use the concept of model discrepancy [76–78] and set prediction accuracy, structural accuracy and parameter accuracy as our performance criteria. A good prediction error does not guarantee the model has good structural accuracy and parameter accuracy, and vice versa, motivating multiple performance criteria.

(b) Numerical experiments

We use the Michaelis–Menten kinetics, given by equation (3.9), to compare the performance of SINDy-PI and implicit-SINDy. We performed our numerical experiments as follows:

Step 1: Randomly generate 2400 different initial conditions of different magnitudes ranging from 0 to 12.5. Simulate those initial conditions using a fourth-order Runge–Kutta method with time step $\mathrm{d}t = 0.1$ and time horizon $T = 5$. The testing data is generated using 600 random initial conditions using the same method as the training data.

Step 2: Add Gaussian noise to the training and testing data. 23 different Gaussian noise levels with magnitudes ranging from 10^{-7} to 5×10^{-1} are used. For each noise level, 30 different random noise realizations are generated, resulting in 30 different noisy datasets for each noise level.

Step 3: Compute the derivative of the noisy data. We investigate several approaches, including finite-difference and total-variation regularized difference (TVRegDiff) [56] derivatives. In all cases, SINDy-PI is several orders of magnitude more robust to noise than implicit-SINDy, and only the result of using TVRegDiff is shown in this paper. TVRegDiff generates more accurate derivatives, but also requires hyperparameter tuning and causes aliasing, so we trim the ends of the time series generated by each initial condition (first and last 30%). It is possible to add Gaussian noise to the clean derivative data to investigate robustness, although this is less relevant for real-world scenarios, where only noisy state data is available.

Step 4: Train SINDy-PI and implicit-SINDy models on noisy training data. For each noise level, we sweep through 68 different sparsity parameters λ for SINDy-PI, from 0.01 to 5. The λ is varied by a factor of 2 [52] to calculate the null space in the implicit-SINDy method. The library for implicit-SINDy and SINDy-PI is

$$\boldsymbol{\Theta}(\mathbf{X}, \dot{\mathbf{X}}) = [\mathbf{1}\ \mathbf{X}\ \mathbf{X}^2\ \mathbf{X}^3\ \mathbf{X}^4\ \dot{\mathbf{X}}\ \dot{\mathbf{X}}\mathbf{X}\ \dot{\mathbf{X}}\mathbf{X}^2\ \dot{\mathbf{X}}\mathbf{X}^3\ \dot{\mathbf{X}}\mathbf{X}^4]. \tag{A 1}$$

Step 5: Due to the various parameter values, we use model selection to choose a model. We use the test data with the same noise magnitude to perform the model selection process. The ratio of training data and testing data is 8 : 2.

Step 6: The best model generated by the two methods is compared. We use the prediction error, error in the model structure (i.e. the number of terms that are incorrectly present or

Table 1. The parameter used for simulating the equation (3.10).

parameter	c_1	c_2	c_3	d_1	d_2	d_3	d_4	e_1	e_2	e_3	e_4	f_1	f_2
value	2.5	−100	13.6769	200	13.6769	−6	−6	6	−64	6	16	64	−13
parameter	f_3	f_4	f_5	g_1	g_2	h_1	h_2	h_3	h_4	h_5	j_1	j_2	j_3
value	13	−16	−100	1.3	−3.1	−200	13.6769	128	−1.28	−32	6	−18	−100

missing from the model), and parameter error as our model performance evaluation criteria. We average the performance over 30 different noise realizations for each noise level. We then plot the distribution of structure error in figure 3.

Many parameters affect the performance of these methods: the length of training data, prediction steps to calculate prediction error, the initial conditions for training data, choice of the library, and the derivative computation. We have attempted to carefully optimize each method, although an exhaustive parameter sweep is beyond the scope of the present work. However, in all cases SINDy-PI outperforms implicit-SINDy.

Appendix B. Data usage of SINDy-PI and implicit-SINDy

Section 3d investigates the data usage of SINDy-PI and implicit-SINDy on the yeast glycolysis model in equation (3.10). The parameters of this problems are given in table 1. The data usage comparison is performed by the following steps:

Step 1: Generate training data by simulating equation (3.10) with parameters in table 1 and a time step of $\mathrm{d}t = 0.1$, with time horizon $T = 5$. We simulate the system using 900 random initial conditions with magnitude ranging from 0 to 3.

Step 2: Shuffle the training data and select j per cent of the entire training dataset at random to train the SINDy-PI and implicit-SINDy models. These training data are sampled from all trajectories, and they are not necessarily consecutive in time. No noise is added since we only care about the effect of the data length in this case. The sparsity parameter λ is fixed for both algorithms (different values); this value is selected for a single percentage j where both methods fail to identify the correct model, and we sweep through λ.

Step 3: Run the numerical experiment for 20 times for each data length and calculate the percentage of times the two algorithms yield the correct structure of the equation (3.10*f*).

The final comparison is shown in figure 4. Data usage requirements for other state equations are given in table 2; figure 4 shows results for the hardest equation to identify. The other equations require less data. Normalizing the SINDy-PI library improves data learning rates as well.

Appendix C. SINDy-PI and PDE-FIND on rational PDE problem

In §3e, we compared the performance of SINDy-PI and PDE-FIND on a modified KdV equation. The simulation data is obtained using a spectral method [74] with a time step of $\mathrm{d}t = 0.01$ and time horizon $T = 20$, spatial domain $L = -25$ to 25, and $n = 128$ spatial discretization points. The library of PDE-FIND is chosen to be

$$\boldsymbol{\Theta}(\mathbf{U}, \mathbf{U}_x, \mathbf{U}_{xx}, \mathbf{U}_{xxx}) = [\mathbf{1}\ \mathbf{U}\ \mathbf{U}_x\ \mathbf{U}_{xx}\ \mathbf{U}_{xxx}\ \mathbf{U}_x^2\ \mathbf{U}_{xx}^2\ \mathbf{U}_{xxx}^2\ \mathbf{U}\mathbf{U}_x\ \mathbf{U}\mathbf{U}_{xx}\ \mathbf{U}\mathbf{U}_{xxx}\ \mathbf{U}^2\mathbf{U}_x^2\ \mathbf{U}^2\mathbf{U}_{xx}^2\ \mathbf{U}^2\mathbf{U}_{xxx}^2] \tag{C 1}$$

Table 2. Comparison of data usage of SINDy-PI and implicit-SINDy on other states.

	equation	equation (3.10*a*)		equation (3.10*b*)		equation (3.10*c*)	equation (3.10*d*)	equation (3.10*e*)	equation (3.10*f*)		equation (3.10*g*)
	library order	6		6		3	3	3	6		3
SINDy-PI un-normalized	left-hand side	$\dot{x}_1$	$\dot{x}_1x_6^4$	$\dot{x}_2$	$\dot{x}_2x_6^4$	$\dot{x}_3$	$\dot{x}_4$	$\dot{x}_5$	$\dot{x}_6x_4^6$		$\dot{x}_7$
	threshold	0.5	0.05	0.5	0.05	0.2	0.5	0.3	0.01		0.5
	data usage	50%	7.5%	55%	8.5%	0.5%	0.5%	0.3%	40%		0.5%
SINDy-PI normalized	left-hand side	$\dot{x}_1$	$\dot{x}_1x_6^4$	$\dot{x}_2$	$\dot{x}_2x_6^4$	$\dot{x}_3$	$\dot{x}_4$	$\dot{x}_5$	$\dot{x}_6$	$\dot{x}_6x_4^6$	$\dot{x}_7$
	threshold	0.5	0.5	0.5	0.5	0.6	0.8	0.4	0.1	0.1	0.2
	data usage	18%	3%	10%	3%	0.45%	0.45%	0.275%	35%	8%	0.4%
implicit-SINDy normalized	threshold	5×10^{-3}		2×10^{-3}		8×10^{-3}	8×10^{-3}	8×10^{-3}	3×10^{-3}		8×10^{-3}
	data usage	10%		10%		0.5%	0.6%	0.3%	100%		0.5%

Table 3. Parameters used to simulate the double pendulum.

parameter	m_1	m_2	L_1	L_2	a_1	a_2	I_1	I_2	g
value	0.2704	0.2056	0.2667	0.2667	0.191	0.1621	0.003	0.0011	9.81

Table 4. Parameters used in equation (4.2) for simulating the Belousov–Zhabotinsky reaction model.

parameter	q	f	ε	α	β	γ	ε_2	ε_3	χ	D_x	D_z	D_s	D_u
value	1	1.5	0.3	0.3	0.26	0.4	0.15	0.03	0	0.01	0.01	1	1

and the right-hand side library for the SINDy-PI is chosen to be

$$\boldsymbol{\Theta}(\mathbf{U},\mathbf{U}_x,\mathbf{U}_{xx},\mathbf{U}_{xxx}) = [\mathbf{1}\ \mathbf{U}\ \mathbf{U}_t\ \mathbf{U}_x\ \mathbf{U}_{xx}\ \mathbf{U}_{xxx}\ \mathbf{U}^2\ \mathbf{U}_t^2\ \mathbf{U}_x^2\ \mathbf{U}_{xx}^2\ \mathbf{U}_{xxx}^2 \\ \mathbf{U}\mathbf{U}_t\ \mathbf{U}\mathbf{U}_x\ \mathbf{U}\mathbf{U}_{xx}\ \mathbf{U}\mathbf{U}_{xxx}\ \mathbf{U}^2\mathbf{U}_x^2\ \mathbf{U}^2\mathbf{U}_{xx}^2\ \mathbf{U}^2\mathbf{U}_{xxx}^2], \tag{C 2}$$

while the left-hand side library is chosen to be

$$C(\mathbf{U},\mathbf{U}_t,\mathbf{U}_x,\mathbf{U}_{xx}) = [\mathbf{U}_t\ \mathbf{U}\mathbf{U}_t\ \mathbf{U}\mathbf{U}_x\ \mathbf{U}\mathbf{U}_{xx}]. \tag{C 3}$$

For both SINDy-PI and PDE-FIND, we used 100 different values for the sparsity parameter λ ranging from 0.1 to 10 with step size 0.1. We use 80% of the simulation data for training and 20% for testing and model selection. We calculate the normalized prediction error for all models on state u_t and the model with minimum prediction error is selected as the final model.

Appendix D. Parameter values for simulations

The parameters for the double pendulum simulation in §4a are given in table 3. The parameters used to simulate the simplified model of the Belousov–Zhabotinsky reaction in equation (4.2) are given in table 4.

Appendix E. SINDy-PI models for the single pendulum on a cart

The Lagrangian for the single pendulum on a cart with an input force on the cart is:

$$\mathcal{L} = T - V = \tfrac{1}{2}(m+M)\dot{s}^2 + \tfrac{1}{2}L^2 m\dot{\phi}^2 - Lgm\cos(\phi) + Lm\cos(\phi)\dot{\phi}\dot{s}, \tag{E 1}$$

where m is the mass at the end of the pendulum arm, M is the mass of the cart, L is the length of the pendulum arm, s is the position of the cart and ϕ is the pendulum angle. We do not consider damping in this case. Using the numeric values $m = M = L = 1$ and $g = -9.81$ this simplifies to

$$\mathcal{L} = T - V = \dot{s}^2 + \tfrac{1}{2}\dot{\phi}^2 - 9.81\cos(\phi) + \cos(\phi)\dot{\phi}\dot{s}. \tag{E 2}$$

The Euler–Lagrange equation of the system are

$$\begin{array}{l} \dfrac{\mathrm{d}}{\mathrm{d}t}\dfrac{\partial\mathcal{L}}{\partial\dot{\phi}} - \dfrac{\partial\mathcal{L}}{\partial\phi} = 0, \\ \dfrac{\mathrm{d}}{\mathrm{d}t}\dfrac{\partial\mathcal{L}}{\partial\dot{s}} - \dfrac{\partial\mathcal{L}}{\partial s} = F, \end{array} \implies \begin{array}{r} mL^2\ddot{\phi} + mL\ddot{s}\cos(\phi) - Lgm\sin(\phi) = 0 \\ (M+m)\ddot{s} - F - mL\sin(\phi)\dot{\phi}^2 + mL\ddot{\phi}\cos(\phi) = 0 \end{array} \tag{E 3a}$$

where F is the force applied to the pendulum cart. It is possible to isolate $\ddot{\phi}$ and $\ddot{s}$:

$$\ddot{\phi} = \frac{-(F\cos(\phi) - Mg\sin(\phi) - mg\sin(\phi) + Lm\cos(\phi)\sin(\phi)\dot{\phi}^2)}{L(M + m\sin(\phi)^2)}, \tag{E 4a}$$

and

$$\ddot{s} = \frac{F + Lm\sin(\phi)\dot{\phi}^2 - mg\cos(\phi)\sin(\phi)}{M + m\sin(\phi)^2}. \tag{E 4b}$$

Table 5. Parameters used to simulate the single pendulum on a cart.

parameter	m	L	M	g
value	1	1	1	9.81

Table 6. Parameters identified by SINDy-PI for equation (E 6*c*) under different noise magnitudes.

basis	numerator			denominator	
value	$\sin(\phi)$	$F\cos(\phi)$	$\sin(\phi)\cos(\phi)\dot{\phi}^2$	constant	$\cos(\phi)^2$
noise magnitude					
0	19.62	−1	−1	2	−1
0.001	19.618	−1.0005	−0.9999	2	−1
0.005	19.6135	−1.171	−0.9996	2	−0.9997
0.02	19.5881	not identified	−0.4912	2	−1.0122

It is possible to write this as a system of four coupled first-order equations

$$\frac{\mathrm{d}}{\mathrm{d}t}\phi = \dot{\phi}, \tag{E 5a}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}s = \dot{s}, \tag{E 5b}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\dot{\phi} = \frac{-(F\cos(\phi) - Mg\sin(\phi) - mg\sin(\phi) + Lm\cos(\phi)\sin(\phi)\dot{\phi}^2)}{L(M + m\sin(\phi)^2)}, \tag{E 5c}$$

and

$$\frac{\mathrm{d}}{\mathrm{d}t}\dot{s} = \frac{F + Lm\sin(\phi)\dot{\phi}^2 - mg\cos(\phi)\sin(\phi)}{M + m\sin(\phi)^2}. \tag{E 5d}$$

With the numerical values shown in table 5, this becomes

$$\frac{\mathrm{d}}{\mathrm{d}t}\phi = \dot{\phi}, \tag{E 6a}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}s = \dot{s}, \tag{E 6b}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\dot{\phi} = \frac{19.62\sin(\phi) - F\cos(\phi) - \sin(\phi)\cos(\phi)\dot{\phi}^2}{2 - \cos(\phi)^2}, \tag{E 6c}$$

and

$$\frac{\mathrm{d}}{\mathrm{d}t}\dot{s} = \frac{2F - 9.81\sin(2\phi) + 2\sin(\phi)\dot{\phi}^2}{2 + 2\sin(\phi)^2}. \tag{E 6d}$$

Parameters identified by SINDy-PI under different noise magnitudes are presented in tables 6 and 7.

Appendix F. SINDy-PI models for the mounted double pendulum

For a mounted double pendulum system shown in figure 6 we could have the following parameters: the parameters of the pendulum are centre of mass m_1 and m_2, centre of mass position a_1 and a_2, arm length L_1 and L_2, arm inertia I_1 and I_2, arm rotational angle ϕ_1 and ϕ_2, gravity acceleration g. Those values could be seen from table 3. If we consider friction between the pendulum joint, we could define $k_1 = 7.2485 \times 10^{-4}$ and $k_2 = 1.6522 \times 10^{-4}$ as our damping

Table 7. Parameters identified by SINDy-PI for equation (E 6*d*) under different noise magnitudes.

basis	numerator			denominator	
value	F	sin (2ϕ)	sin $(\phi)\dot{\phi}^2$	constant	sin $(\phi)^2$
noise magnitude					
0	2	−9.81	2	2	2
0.001	1.9992	−9.816	1.9992	2	1.9992
0.005	1.9982	−9.8015	1.9986	2	1.9986
0.02	2.0705	−9.8234	2.0041	2	2.0041

coefficient. It is easy to derive the Lagrangian of the mounted double pendulum which is given by

$$\begin{aligned}\mathcal{L} = T - V = {} & \frac{(m_2((L_1\cos(\phi_1)\dot{\phi}_1 + a_2\cos(\phi_2)\dot{\phi}_2)^2 + (L_1\sin(\phi_1)\dot{\phi}_1 + a_2\sin(\phi_2)\dot{\phi}_2)^2))}{2} \\ & + \frac{(m_1(a_1^2\cos(\phi_1)^2\dot{\phi}_1^2 + a_1^2\sin(\phi_1)^2\dot{\phi}_1^2))}{2} + \frac{(I_1\dot{\phi}_1^2)}{2} + \frac{(I_2\dot{\phi}_2^2)}{2} - gm_2(a_2\cos(\phi_2) \\ & + L_1\cos(\phi_1)) - a_1 g m_1\cos(\phi_1). \end{aligned} \tag{F 1}$$

The damping term caused by friction with friction coefficients k_1 and k_2 is

$$R_a = \tfrac{1}{2}k_1\dot{\phi}_1 + \tfrac{1}{2}k_2(\dot{\phi}_1 - \dot{\phi}_2)^2. \tag{F 2}$$

The Euler–Lagrange equations with a Rayleigh dissipation term are then:

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial\mathcal{L}}{\partial\dot{\phi}_1} - \frac{\partial\mathcal{L}}{\partial\phi_1} + \frac{\partial R_a}{\partial\dot{\phi}_1} = 0 \tag{F 3a}$$

and

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial\mathcal{L}}{\partial\dot{\phi}_2} - \frac{\partial\mathcal{L}}{\partial\phi_2} + \frac{\partial R_a}{\partial\dot{\phi}_2} = 0. \tag{F 3b}$$

The symbolic form of the equation (F 3*a*) is

$$\begin{aligned} & I_1\ddot{\phi}_1 + k_1\dot{\phi}_1 + k_2\dot{\phi}_1 + L_1^2\ddot{\phi}_1 m_2 + a_1^2\ddot{\phi}_1 m_1 + L_1 a_2 m_2\sin(\phi_1 - \phi_2)\dot{\phi}_2^2 \\ & \quad + L_1 a_2\ddot{\phi}_2 m_2\cos(\phi_1 - \phi_2) - k_2\dot{\phi}_2 - L_1 g m_2\sin(\phi_1) - a_1 g m_1\sin(\phi_1) = 0, \end{aligned} \tag{F 4}$$

and the symbolic form of equation (F 3*b*) is

$$\begin{aligned} & I_2\ddot{\phi}_2 + k_2\dot{\phi}_2 + a_2^2\ddot{\phi}_2 m_2 + L_1 a_2\ddot{\phi}_1 m_2\cos(\phi_1 - \phi_2) - k_2\dot{\phi}_1 \\ & \quad - a_2 g m_2\sin(\phi_2) - L_1 a_2 m_2\sin(\phi_1 - \phi_2)\dot{\phi}_1^2 = 0. \end{aligned} \tag{F 5}$$

Using the numerical parameter values in these equations gives

$$\ddot{\phi}_1 + 0.03235\dot{\phi}_1 + 0.323\ddot{\phi}_2\cos(\phi_1 - \phi_2) + 0.323\dot{\phi}_2^2\sin(\phi_1 - \phi_2) - 0.006006\dot{\phi}_2 - 37.97\sin(\phi_1) = 0.$$

$$\ddot{\phi}_2 + 0.02525\dot{\phi}_2 + 1.358\ddot{\phi}_1\cos(\phi_1 - \phi_2) - 0.02525\dot{\phi}_1 - 49.94\sin(\phi_2) - 1.358\dot{\phi}_1^2\sin(\phi_1 - \phi_2) = 0.$$

If we set $k_1 = k_2 = 0$ and combine the equations, it is possible to solve for $\ddot{\phi}_1$ and $\ddot{\phi}_2$

$$\begin{aligned}\ddot{\phi}_1 = {} & (L_1 a_2^2 g m_2^2\sin(\phi_1) - 2L_1 a_2^3\dot{\phi}_2^2 m_2^2\sin(\phi_1 - \phi_2) + 2I_2 L_1 g m_2\sin(\phi_1) \\ & + L_1 a_2^2 g m_2^2\sin(\phi_1 - 2\phi_2) + 2I_2 a_1 g m_1\sin(\phi_1) - L_1^2 a_2^2\dot{\phi}_1^2 m_2^2\sin(2\phi_1 - 2\phi_2) \\ & - 2I_2 L_1 a_2\dot{\phi}_2^2 m_2\sin(\phi_1 - \phi_2) + 2a_1 a_2^2 g m_1 m_2\sin(\phi_1))/(2I_1 I_2 + L_1^2 a_2^2 m_2^2 \\ & + 2I_2 L_1^2 m_2 + 2I_2 a_1^2 m_1 + 2I_1 a_2^2 m_2 - L_1^2 a_2^2 m_2^2\cos(2\phi_1 - 2\phi_2) + 2a_1^2 a_2^2 m_1 m_2)\end{aligned}$$

and

$$
\begin{aligned}
\ddot{\phi}_2 = {} & (a_2 m_2(2I_1 g \sin(\phi_2) + 2L_1^3 \dot{\phi}_1^2 m_2 \sin(\phi_1 - \phi_2) + 2L_1^2 g m_2 \sin(\phi_2) + 2I_1 L_1 \dot{\phi}_1^2 \sin(\phi_1 - \phi_2) \\
& + 2a_1^2 g m_1 \sin(\phi_2) + L_1^2 a_2 \dot{\phi}_2^2 m_2 \sin(2\phi_1 - 2\phi_2) + 2L_1 a_1^2 \dot{\phi}_1^2 m_1 \sin(\phi_1 - \phi_2) \\
& - 2L_1^2 g m_2 \cos(\phi_1 - \phi_2)\sin(\phi_1) - 2L_1 a_1 g m_1 \cos(\phi_1 - \phi_2)\sin(\phi_1)))/(2(I_1 I_2 \\
& + L_1^2 a_2^2 m_2^2 + I_2 L_1^2 m_2 + I_2 a_1^2 m_1 + I_1 a_2^2 m_2 - L_1^2 a_2^2 m_2^2 \cos(\phi_1 - \phi_2)^2 + a_1^2 a_2^2 m_1 m_2)).
\end{aligned}
$$

If we use the values in table 3 we have

$$
\begin{aligned}
\ddot{\phi}_1 = {} & (-0.2808 \sin(2\phi_1 - 2\phi_2)\dot{\phi}_1^2 - 0.4136 \sin(\phi_1 - \phi_2)\dot{\phi}_2^2 \\
& + 10.3278 \sin(\phi_1 - 2\phi_2) + 38.2984 \sin(\phi_1))/(1 - 0.2808 \cos(2\phi_1 - 2\phi_2)),
\end{aligned}
$$

and

$$
\begin{aligned}
\ddot{\phi}_2 = {} & (1.7390 \sin(\phi_1 - \phi_2)\dot{\phi}_1^2 + 0.2808 \sin(2\phi_1 - 2\phi_2)\dot{\phi}_2^2 \\
& - 33.02 \sin(2\phi_1 - \phi_2) + 30.9472 \sin(\phi_2))/(1 - 0.2808 \cos(2\phi_1 - 2\phi_2)).
\end{aligned}
$$

With no noise, SINDy-PI discovers the correct equations. When we add random noise with magnitude of 0.005, SINDy-PI discovers the following

$$
\begin{aligned}
\ddot{\phi}_1 = {} & (-0.2799 \sin(2\phi_1 - 2\phi_2)\dot{\phi}_1^2 - 0.4137 \sin(\phi_1 - \phi_2)\dot{\phi}_2^2 \\
& + 10.3429 \sin(\phi_1 - 2\phi_2) + 38.3117 \sin(\phi_1))/(1 - 0.2815 \cos(2\phi_1 - 2\phi_2)),
\end{aligned}
$$

and

$$
\begin{aligned}
\ddot{\phi}_2 = {} & (1.7392 \sin(\phi_1 - \phi_2)\dot{\phi}_1^2 + 0.2805 \sin(2\phi_1 - 2\phi_2)\dot{\phi}_2^2 \\
& - 33.0035 \sin(2\phi_1 - \phi_2) + 30.9418 \sin(\phi_2))/(1 - 0.2813 \cos(2\phi_1 - 2\phi_2)).
\end{aligned}
$$

If we increase the noise magnitude to 0.01 then the SINDy-PI discovered equation becomes

$$
\begin{aligned}
\ddot{\phi}_1 = {} & (-0.2768\dot{\phi}_1^2 \sin(2\phi_1 - 2\phi_2) - 0.4138 \sin(\phi_1 - \phi_2)\dot{\phi}_2^2 \\
& + 10.3676 \sin(\phi_1 - 2\phi_2) + 38.3225 \sin(\phi_1))/(1 - 0.2818 \cos(2\phi_1 - 2\phi_2)),
\end{aligned}
$$

and

$$
\begin{aligned}
\ddot{\phi}_2 = {} & (1.7355 \sin(\phi_1 - \phi_2)\dot{\phi}_1^2 + 0.2794 \sin(2\phi_1 - 2\phi_2)\dot{\phi}_2^2 + 0.1675 \sin(2\phi_1 - 2\phi_2) \\
& - 33.0445 \sin(2\phi_1 - \phi_2) + 31.0065 \sin(\phi_2))/(1 - 0.2819 \cos(2\phi_1 - 2\phi_2)).
\end{aligned}
$$

If we continue to increase the noise magnitude to 0.05 then SINDy-PI incorrectly identifies

$$
\begin{aligned}
\ddot{\phi}_1 = {} & (15.5413 \sin(\phi_1 - 2\phi_2) - 2.6396 \sin(\phi_1 - \phi_2) - 0.9538 \cos(\phi_1 - \phi_2) \\
& + 35.9971 \cos(\phi_1 - 61/40) - 2.4160 \cos(\phi_2 - 1149/1000) - 0.0733 \cos(2\phi_1 - 2\phi_2) \\
& + 0.0269 \cos(4\phi_1 - 2\phi_2) + 2.3419 \sin(2\phi_1 - \phi_2) + 0.5142\dot{\phi}_1 \sin(2\phi_1 - 2\phi_2) \\
& - 0.4584\dot{\phi}_2 \sin(2\phi_1 - \phi_2) - 0.3807\dot{\phi}_2^2 \sin(\phi_1 - \phi_2) + 0.8810\dot{\phi}_1 \sin(\phi_1 - \phi_2) \\
& + 0.9411\dot{\phi}_1 \sin(\phi_1 - 2\phi_2) - 0.7664\dot{\phi}_2 \sin(\phi_1 - \phi_2) + 1.2026)/(1 - 0.4245 \cos(2\phi_1 - 2\phi_2)),
\end{aligned}
$$

and

$$
\ddot{\phi}_2 = 70.9 \sin(\phi_1 - \phi_2).
$$

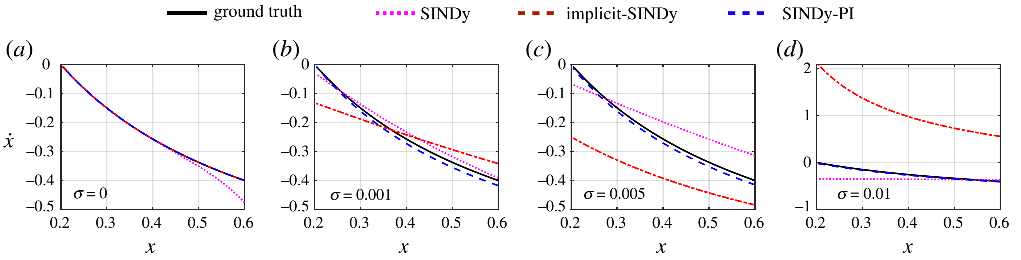


Figure 10. Comparison of the model identified by SINDy, Implicit-SINDy, and SINDy-PI on Michaelis–Menten dynamics. (*a*) Both SINDy-PI and Implicit-SINDy identified the correct model. However, the SINDy model only agrees for *x* near the origin. (*b*–*d*) As Gaussian noise level increases, the SINDy model degrades further. (Online version in colour.)

Appendix G. SINDy-PI model for the Belousov–Zhabotinsky reaction

The SINDy-PI discovered PDE for the simplified BZ reaction is

$$
\begin{aligned}
x_\tau &= \Delta x + \frac{0.24667x + 0.33333s + 0.5z + 3.3333xs - 5.0xz + 2.1333x^2 - 3.3333x^3}{x + 0.1},\\
z_\tau &= 0.01\Delta z + x + 0.4u - 1.3z,\\
s_\tau &= \Delta s + 0.17333r - 0.66667s,\\
u_\tau &= \Delta u - 133.33u + 100z.
\end{aligned}
$$

Appendix H. Inability to identify rational dynamics with SINDy

In this section, we demonstrate that it is not possible to identify rational dynamics with the original SINDy algorithm, testing it on the Michaelis–Menten dynamics in equation (3.9). We use the same parameters in §3c. The Taylor expansion of equation (3.9) at $x = 0$ is

$$\dot{x} \approx 0.6 - 5x + \frac{50}{3}x^2 - \frac{500}{9}x^3 + \frac{5000}{27}x^4 - \frac{50000}{81}x^5. \tag{H 1}$$

Thus, when the trajectory provided for training is close enough to $x = 0$, SINDy should identify equation (H 1). To verify this, data with $x_0 = 0.2409$ is simulated for 22 time steps with $\mathrm{d}t = 0.01$. Both the Implicit-SINDy and SINDy-PI algorithms identify the correct model in equation (3.9) with highly accurate parameters. The model identified by SINDy is

$$\dot{x} = 0.5914 - 4.7387x + 13.1389x^2 - 27.6470x^3 + 36.9846x^4 - 22.8388x^5. \tag{H 2}$$

Thus, SINDy correctly identifies the first three terms of the Taylor expansion, although the higher order terms have large parameter errors. This model is compared with the SINDy-PI and Implicit-SINDy models in figure 10 for a test trajectory initialized with $x_0 = 0.6$. From figure 10*a*, it can be seen that both SINDy-PI and Implicit-SINDy match the true solution. However, the SINDy model only agrees for x near the origin. When Gaussian noise of magnitude σ is added, the SINDy model degrades further. Moreover, the amount of data used needs to be increased to identify the correct model. In figure 10*b*–*d*, 330, 2200 and 4400 data points ranging from 0 to 12 are used for training. The same amount of data is used for model selection. From figure 10*b*–*d*, it can be seen that the SINDy model does not work well away from $x = 0$.

Appendix I. Robustness of SINDy and SINDy-PI

The robustness of SINDy-PI to noise depends on a number of factors, including the the length of the data, which initial conditions are chosen to generate the data, the model parameters, the order

Table 8. The effect of K_m on the noise robustness of SINDy-PI.

K_m	max magnitude of noise	true model	identified model
0.01	0.001	$\dot{x} = -\frac{(0.9x - 0.006)}{x + 0.01}$	$\dot{x} = -\frac{(x - 7.6144)(x - 5.6092)(x - 2.5078)(x - 0.0072)}{(x - 7.6146)(x - 5.6125)(x - 2.5092)(x + 0.0092)}$
0.1	0.01	$\dot{x} = -\frac{(0.9x - 0.06)}{x + 0.1}$	$\dot{x} = -\frac{(x - 5.4132)(x - 0.0681)}{(x - 5.4144)(x + 0.099)}$
1	0.05	$\dot{x} = -\frac{(0.9x - 0.6)}{x + 1}$	$\dot{x} = -\frac{(x - 4.7143)(x - 0.6726)}{(x - 4.7159)(x + 0.9698)}$
10	0.1	$\dot{x} = -\frac{(0.9x - 6)}{x + 10}$	$\dot{x} = -\frac{0.99x - 6.583}{x + 11.07}$

of the polynomial terms in the model, etc. The relative impact of all of these factors varies with the system we are studying.

In general, training data that explores more of the phase space will result in a more robust model discovery process. Generally speaking, data that results in a better-conditioned $\boldsymbol{\Theta}$ matrix will provide more robust results. Several studies have explored strategies to improve the robustness, with Wu and Xiu suggesting the use of a large ensemble of initial conditions [43]. Exciting transients is also important [28].

Another key factor that affects the condition number of $\boldsymbol{\Theta}$ is the number of library elements, which is determined by the order for a polynomial library. Including higher-order terms increases the condition number, making it more difficult to accurately disambiguate which nonlinear term is responsible for the observed behaviour. The library size scales exponentially with the maximum polynomial order.

Noise robustness is also affected by the model parameters. Smaller parameters in $\boldsymbol{\Xi}$ are more likely to be removed during thresholding, making the procedure less robust to noise. To illustrate this, we change the parameter K_m in (3.9) and test the maximum noise SINDy-PI can handle. We set K_m equal to $0.01, 0.1, 1$ and 10 separately. We generate the training data with 120 random initial conditions ranging from 0 to 10. Each initial condition is simulated until $T = 5$ with $\mathrm{d}t = 0.01$, and the same magnitude of Gaussian noise is added. TVRegDiff is used to calculate the derivative, and the first and last 30% of data is discarded due to aliasing effect. The testing data is generated using the same process and the ratio of training and testing data is $1:1$. The model and maximum magnitude of noise allowed for each values of K_m is summarized in table 8. These results suggest that as K_m increases, the maximum noise SINDy-PI can handle also increases.

References

1. Brunton SL, Kutz JN. 2019 *Data-driven science and engineering: machine learning, dynamical systems, and control*. Cambridge UK: Cambridge University Press.
2. Nelles O. 2013 *Nonlinear system identification: from classical approaches to neural networks and fuzzy models*. Berlin, Germany: Springer.
3. Ljung L. 2010 Perspectives on system identification. *Annu. Rev. Control* **34**, 1–12. (doi:10.1016/j.arcontrol.2009.12.001)
4. Schmid PJ. 2010 Dynamic mode decomposition of numerical and experimental data. *J. Fluid Mech.* **656**, 5–28. (doi:10.1017/S0022112010001217)
5. Kutz JN, Brunton SL, Brunton BW, Proctor JL. 2016 *Dynamic mode decomposition: data-driven modeling of complex systems*. Philadelphia, PA: SIAM.
6. Budišić M, Mohr R, Mezić I. 2012 Applied Koopmanism. *Chaos* **22**, 047510. (doi:10.1063/1.4772195)
7. Mezic I. 2013 Analysis of fluid flows via spectral properties of the Koopman operator. *Annu. Rev. Fluid Mech.* **45**, 357–378. (doi:10.1146/annurev-fluid-011212-140652)
8. Williams MO, Kevrekidis IG, Rowley CW. 2015 A data-driven approximation of the Koopman operator: extending dynamic mode decomposition. *J. Nonlinear Sci.* **6**, 1307–1346. (doi:10.1007/s00332-015-9258-5)
9. Klus S, Nüske F, Koltai P, Wu H, Kevrekidis I, Schütte C, Noé F. 2018 Data-driven model reduction and transfer operator approximation. *J. Nonlinear Sci.* **28**, 985–1010. (doi:/10.1007/s00332-017-9437-7)

10. Akaike H. 1969 Fitting autoregressive models for prediction. *Ann. Inst. Stat. Math.* **21**, 243–247. (doi:10.1007/BF02532251)
11. Billings SA. 2013 *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains*. Hoboken, NJ: John Wiley & Sons.
12. Yang L, Zhang D, Karniadakis GE. 2018 Physics-informed generative adversarial networks for stochastic differential equations. (http://arxiv.org/abs/1811.02033)
13. Wehmeyer C, Noé F. 2018 Time-lagged autoencoders: deep learning of slow collective variables for molecular kinetics. *J. Chem. Phys.* **148**, 1–9. (doi:10.1063/1.5011399)
14. Mardt A, Pasquali L, Wu H, Noé F. 2018 VAMPnets: deep learning of molecular kinetics. *Nat. Commun.* **9**, 5 pp. (doi:10.1038/s41467-017-02388-1)
15. Vlachas PR, Byeon W, Wan ZY, Sapsis TP, Koumoutsakos P. 2018 Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks. *Proc. R. Soc. A* **474**, 20170844. (doi:10.1098/rspa.2017.0844)
16. Pathak J, Hunt B, Girvan M, Lu Z, Ott E. 2018 Model-free prediction of large spatiotemporally chaotic systems from data: a reservoir computing approach. *Phys. Rev. Lett.* **120**, 024102. (doi:10.1103/PhysRevLett.120.024102)
17. Lu L, Meng X, Mao Z, Karniadakis GE. 2019 DeepXDE: a deep learning library for solving differential equations. (http://arxiv.org/abs/1907.04502)
18. Raissi M, Perdikaris P, Karniadakis G. 2019 Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **378**, 686–707. (doi:10.1016/j.jcp.2018.10.045)
19. Champion K, Lusch B, Kutz JN, Brunton SL. 2019 Data-driven discovery of coordinates and governing equations. *Proc. Natl Acad. Sci. USA* **116**, 22 445–22 451. (doi:10.1073/pnas.1906995116)
20. Raissi M, Yazdani A, Karniadakis GE. 2020 Hidden fluid mechanics: learning velocity and pressure fields from flow visualizations. *Science* **367**, 1026–1030. (doi:10.1126/science.aaw4741)
21. Raissi M, Karniadakis GE. 2017 Machine learning of linear differential equations using Gaussian processes. (http://arxiv.org/abs/1701.02440)
22. Raissi M, Karniadakis GE. 2018 Hidden physics models: machine learning of nonlinear partial differential equations. *J. Comput. Phys.* **357**, 125–141. (doi:10.1016/j.jcp.2017.11.039)
23. Giannakis D, Majda AJ. 2012 Nonlinear Laplacian spectral analysis for time series with intermittency and low-frequency variability. *Proc. Natl Acad. Sci. USA* **109**, 2222–2227. (doi:10.1073/pnas.1118984109)
24. Yair O, Talmon R, Coifman RR, Kevrekidis IG. 2017 Reconstruction of normal forms by learning informed observation geometries from data. *Proc. Natl Acad. Sci. USA* **114**, E7865–E7874. (doi:10.1073/pnas.1620045114)
25. Bongard J, Lipson H. 2007 Automated reverse engineering of nonlinear dynamical systems. *Proc. Natl Acad. Sci. USA* **104**, 9943–9948. (doi:10.1073/pnas.0609476104)
26. Schmidt M, Lipson H. 2009 Distilling free-form natural laws from experimental data. *Science* **324**, 81–85. (doi:10.1126/science.1165893)
27. Daniels BC, Nemenman I. 2015 Automated adaptive inference of phenomenological dynamical models. *Nat. Commun.* **6**, 1–8. (doi:10.1038/ncomms9133)
28. Brunton SL, Proctor JL, Kutz JN. 2016 Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proc. Natl Acad. Sci. USA* **113**, 3932–3937. (doi:10.1073/pnas.1517384113)
29. Rudy SH, Brunton SL, Proctor JL, Kutz JN. 2017 Data-driven discovery of partial differential equations. *Sci. Adv.* **3**, e1602614. (doi:10.1126/sciadv.1602614)
30. Schaeffer H. 2017 Learning partial differential equations via data discovery and sparse optimization. *Proc. R. Soc. A* **473**, 20160446. (doi:10.1098/rspa.2016.0446)
31. Sorokina M, Sygletos S, Turitsyn S. 2016 Sparse identification for nonlinear optical communication systems: SINO method. *Opt. Express* **24**, 30 433–30 443. (doi:10.1364/OE.24.030433)
32. Dam M, Brøns M, Juul Rasmussen J, Naulin V, Hesthaven JS. 2017 Sparse identification of a predator-prey system from simulation data of a convection model. *Phys. Plasmas* **24**, 022310. (doi:10.1063/1.4977057)
33. Narasingam A, Kwon JS-I. 2018 Data-driven identification of interpretable reduced-order models using sparse regression. *Comput. Chem. Eng.* **119**, 101–111. (doi:10.1016/j.compchemeng.2018.08.010)

34. Loiseau J-C, Brunton SL. 2018 Constrained sparse Galerkin regression. *J. Fluid Mech.* **838**, 42–67. (doi:10.1017/jfm.2017.823)
35. Quade M, Abel M, Kutz JN, Brunton SL. 2018 Sparse identification of nonlinear dynamics for rapid model recovery. *Chaos* **28**, 063116. (doi:10.1063/1.5027470)
36. Boninsegna L, Nüske F, Clementi C. 2018 Sparse learning of stochastic dynamical equations. *J. Chem. Phys.* **148**, 241723. (doi:10.1063/1.5018409)
37. Loiseau J-C, Noack BR, Brunton SL. 2018 Sparse reduced-order modelling: sensor-based dynamics to full-state estimation. *J. Fluid Mech.* **844**, 459–490. (doi:10.1017/jfm.2018.147)
38. Zhang L, Schaeffer H. 2018 On the convergence of the SINDy algorithm. (http://arxiv.org/abs/1805.06445)
39. Hoffmann M, Fröhner C, Noé F. 2019 Reactive SINDy: discovering governing reactions from concentration data. *J. Chem. Phys.* **150**, 025101. (doi:10.1063/1.5066099)
40. Mangan NM, Askham T, Brunton SL, Kutz JN, Proctor JL. 2019 Model selection for hybrid dynamical systems via sparse regression. *Proc. R. Soc. A* **475**, 20180534. (doi:10.1098/rspa.2018.0534)
41. Thaler S, Paehler L, Adams NA. 2019 Sparse identification of truncation errors. *J. Comput. Phys.* **397**, 108851. (doi:10.1016/j.jcp.2019.07.049)
42. Lai Z, Nagarajaiah S. 2019 Sparse structural system identification method for nonlinear dynamic systems with hysteresis/inelastic behavior. *Mech. Syst. Signal Process.* **117**, 813–842. (doi:10.1016/j.ymssp.2018.08.033)
43. Wu K, Xiu D. 2019 Numerical aspects for approximating governing equations using data. *J. Comput. Phys.* **384**, 200–221. (doi:10.1016/j.jcp.2019.01.030)
44. de Silva B, Higdon DM, Brunton SL, Kutz JN. 2019 Discovery of physics from data: universal laws and discrepancies. (http://arxiv.org/abs/1906.07906)
45. Gelß P, Klus S, Eisert J, Schütte C. 2019 Multidimensional approximation of nonlinear dynamical systems. *J. Comput. Nonlinear Dyn.* **14**, 061006. (doi:10.1115/1.4043148)
46. Zhang S, Lin G. 2018 Robust data-driven discovery of governing physical laws with error bars. *Proc. R. Soc. A* **474**, 20180305. (doi:10.1098/rspa.2018.0305)
47. Mangan NM, Brunton SL, Proctor JL, Kutz JN. 2016 Inferring biological networks by sparse identification of nonlinear dynamics. *IEEE Trans. Mol. Biol. Multiscale Commun.* **2**, 52–63. (doi:10.1109/TMBMC.2016.2633265)
48. Kaiser E, Kutz JN, Brunton SL. 2018 Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. *Proc. R. Soc. A* **474**, 20180335. (doi:10.1098/rspa.2018.0335)
49. Schaeffer H, McCalla SG. 2017 Sparse model selection via integral terms. *Phys. Rev. E* **96**, 023302. (doi:10.1103/PhysRevE.96.023302)
50. Reinbold PA, Gurevich DR, Grigoriev RO. 2020 Using noisy or incomplete data to discover models of spatiotemporal dynamics. *Phys. Rev. E* **101**, 010203. (doi:10.1103/PhysRevE.101.010203)
51. Wright J, Yang A, Ganesh A, Sastry S, Ma Y. 2009 Robust face recognition via sparse representation. *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)* **31**, 210–227. (doi:10.1109/TPAMI.2008.79)
52. Qu Q, Sun J, Wright J. 2014 Finding a sparse vector in a subspace: linear sparsity using alternating directions. In *Proc. of the 27th Int. Conf. on Neural Information Processing Systems*, vol. 2, ser. NIPS'14, pp. 3401–3409. Cambridge, MA: MIT Press.
53. Zhu Q, Wang Y, Zhao D, Li S, Billings SA. 2015 Review of rational (total) nonlinear dynamic system modelling, identification, and control. *Int. J. Syst. Sci.* **46**, 2122–2133. (doi:10.1080/00207721.2013.849774)
54. Zhu Q, Liu L, Zhang W, Li S. 2018 Control of complex nonlinear dynamic rational systems. *Complexity* **2018**, 8953035. (doi:10.1155/2018/8953035)
55. Koryakovskiy I, Kudruss M, Vallery H, Babuška R, Caarls W. 2018 Model-plant mismatch compensation using reinforcement learning. *IEEE Robot. Autom. Lett.* **3**, 2471–2477. (doi:10.1109/LRA.2018.2800106)
56. Chartrand R. 2011 Numerical differentiation of noisy, nonsmooth data. *ISRN Appl. Math.* **2011**, 164564. (doi:10.5402/2011/164564)
57. Zhang L, Schaeffer H. 2019 On the convergence of the SINDy algorithm. *Multiscale Model. Simul.* **17**, 948–972. (doi:10.1137/18M1189828)
58. Tibshirani R. 1996 Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B (Methodological)* **58**, 267–288. (doi:10.1111/j.2517-6161.1996.tb02080.x)

59. Zheng P, Askham T, Brunton SL, Kutz JN, Aravkin AY. 2018 A unified framework for sparse relaxed regularized regression: SR3. *IEEE Access* **7**, 1404–1423. (doi:10.1109/ACCESS.2018.2886528)

60. Champion K, Zheng A, Aravkin Peng, Brunton S, Nathan Kutz J. 2019 A unified sparse optimization framework to learn parsimonious physics-informed models from data. (http://arxiv.org/abs/1906.10612). p. 1906.10612v1.

61. Pan W, Yuan Y, Gonçalves J, Stan G. 2016 A sparse Bayesian approach to the identification of nonlinear state-space systems. *IEEE Trans. Autom. Control* **61**, 182–187. (doi:10.1109/TAC.2015.2426291)

62. Gavish M, Donoho DL. 2014 The optimal hard threshold for singular values is $4/\sqrt{3}$. *IEEE Trans. Inf. Theory* **60**, 5040–5053. (doi:10.1109/TIT.2014.2323359)

63. Akaike H. 1998 Information theory and an extension of the maximum likelihood principle. In *Selected papers of Hirotugu Akaike* (eds E Purzen, K Tanabe, G Kitagawa), pp. 199–213. Springer Series in Statistics (Perspectives in Statistics). New York, NY: Springer.

64. Akaike H. 1974 A new look at the statistical model identification. *IEEE Trans. Autom. Control* **19**, 716–723. (doi:10.1109/TAC.1974.1100705)

65. Schwarz G. 1978 Estimating the dimension of a model. *Ann. Stat.* **6**, 461–464. (doi:10.1214/aos/1176344136)

66. Mangan NM, Kutz JN, Brunton SL, Proctor JL. 2017 Model selection for dynamical systems via sparse regression and information criteria. *Proc. R. Soc. A* **473**, 20170009. (doi:10.1098/rspa.2017.0009)

67. Grant M, Boyd S, Ye Y. 2009 CVX: Matlab software for disciplined convex programming, version 2.1, http://evxr.com/evx. March 2014.

68. Grant MC, Boyd SP. 2008 Graph implementations for nonsmooth convex programs. In *Recent advances in learning and control* (eds VD Blondel, SP Boyd, H Kimura), pp. 95–110. Lecture Notes in Control and Information Sciences, vol. 371. London, UK: Springer.

69. Johnson KA, Goody RS. 2011 The original Michaelis constant: translation of the 1913 Michaelis–Menten paper. *Biochemistry* **50**, 8264–8269. (doi:10.1021/bi201284u)

70. Cornish-Bowden A. 2015 One hundred years of Michaelis–Menten kinetics. *Perspect. Sci.* **4**, 3–9. (doi:10.1016/j.pisc.2014.12.002)

71. Schmidt MD, Vallabhajosyula RR, Jenkins JW, Hood JE, Soni AS, Wikswo JP, Lipson H. 2011 Automated refinement and inference of analytical models for metabolic networks. *Phys. Biol.* **8**, 055011. (doi:10.1088/1478-3975/8/5/055011)

72. Graichen K, Treuer M, Zeitz M. 2007 Swing-up of the double pendulum on a cart by feedforward and feedback control with experimental validation. *Automatica* **43**, 63–71. (doi:10.1016/j.automatica.2006.07.023)

73. Vanag VK. 2004 Waves and patterns in reaction–diffusion systems. Belousov–Zhabotinsky reaction in water-in-oil microemulsions. *Phys.-Usp.* **47**, 923. (doi:10.1070/PU2004v047n09ABEH001742)

74. Spectral Methods in MATLAB. SIAM, 2000, vol. 10, ch. 8. Chebyshev Series and the FFT, pp. 75–86. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/1.9780898719598.ch8.

75. Kutz JN. 2013 *Data-driven modeling & scientific computation: methods for complex systems & big data*. New York, NY: Oxford University Press.

76. Kaheman K, Kaiser E, Strom B, Kutz JN, Brunton SL. 2019 Learning discrepancy models from experimental data. In *58th IEEE Conf. on Decision and Control. Nice, France, 11–13 December*, pp. 7389–7396. Piscataway, NJ: IEEE.

77. Kennedy MC, O'Hagan A. 2001 Bayesian calibration of computer models. *J. R. Stat. Soc. Series B Stat. Methodol.)* **63**, 425–464. (doi:10.1111/1467-9868.00294)

78. Arendt PD, Apley DW, Chen W. 2012 Quantification of model uncertainty: calibration, model discrepancy, and identifiability. *J. Mech. Des.* **134**, 100908. (doi:/10.1115/1.4007390)